POLITÉCNICO DE COLOMBIA
educación sin límites

Diplomado en:

Programación en PHP

Guía didáctica Nº 4



Formación Virtual

educación sin límites

GUÍA DIDÁCTICA N°4
M2-DV59-GU04
MÓDULO 4: DESARROLLO WEB I
DIPLOMADO EN PROGRAMACIÓN EN PHP

© DERECHOS RESERVADOS - POLITÉCNICO DE COLOMBIA, 2019
Medellín, Colombia

Proceso: Gestión Académica
Realización del texto: Diego Palacio, Docente
Revisión del texto: - Jehison Posada, Asesor Gramatical
Diseño: Cristian Quintero, Diseñador Gráfico

Editado por el Politécnico de Colombia

ÍNDICE

PRESENTACIÓN 4

COMPETENCIAS 5

TEMA 1 Desarrollo Web 6

TEMA 2 HTML 8

TEMA 3 ENCABEZADOS 12

TEMA 4 ENLACES - HIPERVINCULOS 14

TEMA 5 IMAGENES 17

TEMA 6 TABLAS 20

TEMA 7 FORMULARIOS 26

TEMA 8 HTML - PHP 55

TEMA 9 Bases de Datos - MySQL 85

ASPECTOS CLAVES 92

PRESENTACIÓN

La Guía Didáctica N°4 del MÓDULO 4: DESARROLLO WEB I, es un material que ha sido desarrollado para el apoyo y orientación del participante en el *Diplomado en Programación en PHP*, especialmente, está orientada a la consolidación y/o desarrollo de las habilidades y destrezas necesarias para generar unas adecuadas bases en lo que concierne a la programación en PHP y más específicamente la Programación Web.

Como bien conoces, el objetivo principal de este módulo número 4 es introducir al estudiante en todo a lo referente a Programación Web con PHP, todos sus componentes y conceptos básicos para un adecuado desarrollo idóneo de las bases del desarrollo web.

Para ello, se ha organizado esta guía seis (4) contenidos temáticos, entre ellos: (a) desarrollo web, (b) HTML, (c) HTML y PHP, y (d) Bases de Datos, que ayudarán a formar unas adecuadas bases en todo lo que gira entorno a la programación web con PHP.

COMPETENCIAS

Se espera que con los temas abordados en la Guía Didáctica N°4 del MÓDULO 4: DESARROLLO WEB I, el estudiante logre la siguiente competencia:



- Conocer los conceptos básicos de la programación web en PHP.

Indicadores de logro:

- Comprende la programación web con PHP.
- Conoce la aplicación y funcionamiento de HTML.
- Aplica correctamente todas y cada una de las etiquetas de HTML.
- Vincula PHP y HTML con facilidad.
- Envía datos através de formularios de forma correcta.
- Diferencia y conoce los métodos POST y GET.
- Entiende el significado y uso de phpMyAdmin.
- Crea una base de datos con facilidad.

TEMA 1
Desarrollo Web

PHP es el lenguaje de programación 100% para el desarrollo del web del lado del backend más popular, pero ¿qué es el backend?

Backend

Backend es un término que se usa para referirse a un ámbito de la programación de aplicaciones y software, es la parte que se encarga del acceso y manipulación a los datos y la seguridad de la información. Aunque se usa mucho para referirnos a la programación web que se realiza del lado del servidor, es válido hablar de backend para el desarrollo de todo tipo de aplicaciones.

Con PHP puedes desarrollar sitios y aplicaciones de todo tipo. PHP es el motor además de los CMS más populares, como WordPress, Joomla, Drupal o Magento, entre muchos otros.

Una de las ventajas de PHP es su facilidad de uso. Cualquier persona con muy pocos conocimientos puede comenzar a usar PHP para el desarrollo de páginas web. Sin embargo, esa misma ventaja también puede ser un inconveniente, pues si no se dispone de la debida formación es fácil caer en malas prácticas.

Otra gran ventaja es su disponibilidad, ya que lo encontramos en la mayoría de alojamientos web de todo tipo.

Al hablar de desarrollo web con PHP debemos hablar inmediatamente de HTML y la estrecha relación que existe entre estos.

Todo el mundo de la programación web se sienta sobre las bases de lo que es HTML, toda página, componente o estructura web debe estar ceñida sobre HTML. PHP nos facilita la vida gracias a que el vínculo programacional que existe entre ambos permite crear un dinamismo muy potente en el desarrollo web bajo PHP y haciendo uso de HTML.

PHP se integra con mucha facilidad en páginas HTML y no por nada es uno de los lenguajes más utilizado y soportados en el desarrollo web, más adelante en esta guía entenderá y conocerá la forma en qué, PHP y HTML se dan las manos en el desarrollo.

Los datos también juegan un papel clave en el desarrollo de software web. Cada lenguaje de programación debe estar diseñado para soportar los motores de bases de datos más populares en la actualidad, SQL, MySQL PostgreSQL, etc; PHP en este caso tiene una estrecha relación directa con MySQL dado que viene integrado dentro del motor de ejecución del mismo (XAMPP) y permite administrar una base de datos desde su creación hasta sus copias de seguridad con increíble facilidad.

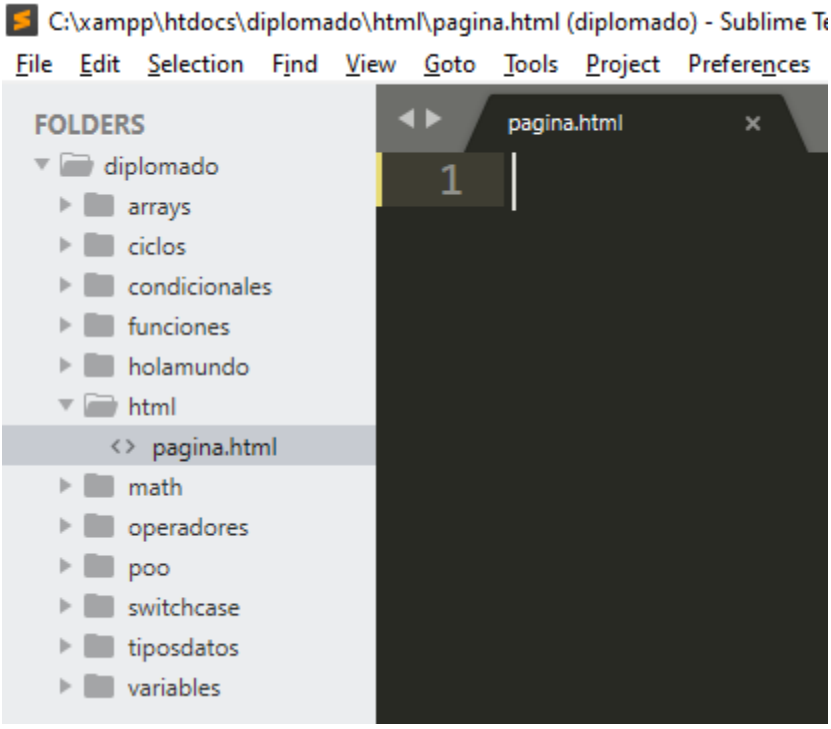
Ahora, ya con la estructura de un sitio web (HTML), un lenguaje de programación potente (PHP) y un motor de bases de datos funciona e integrador con PHP (MySQL) simplemente hay que darle rindas la imaginación, al código y al futuro.

TEMA 2
HTML

En pocas palabras HTML, por sus siglas en inglés de HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web.

- HTML significa lenguaje de marcado de hipertexto
- HTML describe la estructura de una página web
- HTML consiste en una serie de elementos
- Los elementos HTML le dicen al navegador cómo mostrar el contenido
- Los elementos HTML están representados por etiquetas
- Las etiquetas HTML etiquetan piezas de contenido como "encabezado", "párrafo", "tabla", etc.
- Los navegadores no muestran las etiquetas HTML, pero las usan para representar el contenido de la página.

Normalmente HTML se presenta por archivo con la extensión .html y al ser un lenguaje de marcado independiente de PHP no hace falta de XAMPP o Apache (Mientras no se escriba código PHP), un ejemplo puede ser el siguiente. Crear un nuevo folder dentro del proyecto diplomado llamado: "html", en este caso trabajaremos con un solo archivo de extensión .html y se llamará "pagina.html", de la siguiente forma:



HTML es un poco extenso de explicar, pero nos centramos principalmente en los conceptos y características que utilizaremos en el diplomado como lo son:

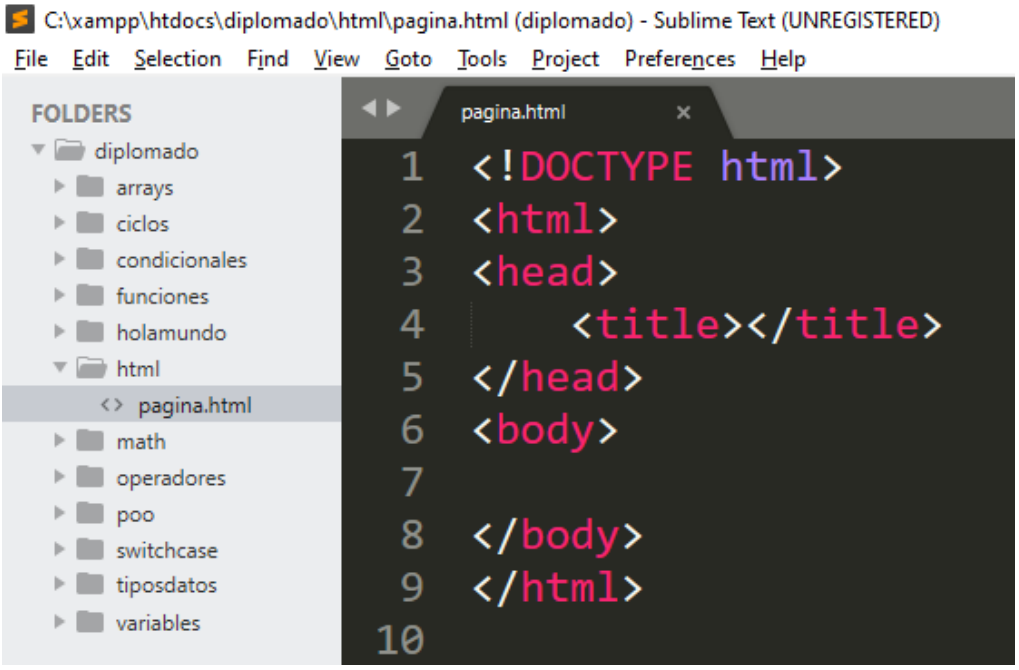
- Elementos básicos
- Tablas
- Formularios
- Botones
- Hipervínculos
- Imágenes
- Atributos y etiquetas

Iniciemos creando nuestra primara página y mostremos algo básico, el "Hola Mundo":

Toda página en HTML tiene unas etiquetas base que se encargan de dividir el contenido de la misma. Estas son:

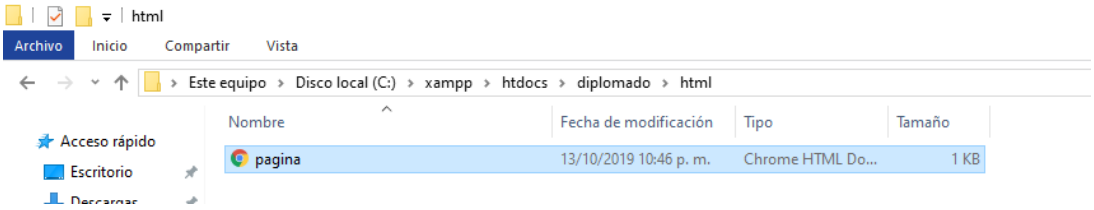
- <!DOCTYPE html>
- <html>
- <head>
- <title>
- <body>

En el código se vería de la siguiente forma:



Existe algo particular con HTML y es que toda etiqueta, tiene una etiqueta de cierre. <> </>. Esto permite delimitar el alcance de un bloque de código. Veamos como se ve la página en el navegador simplemente con este formato.

Para observar el contenido de esta, sin usar XAMMP o Apache, debemos es usar la dirección exacta del archivo o más fácil aún, buscando el mismo el nuestro computador:



Y simplemente abrirlo con doble click. O yendo directamente a la URL en el navegador: file:///C:/xampp/htdocs/diplomado/html/pagina.html.

Recuerda que no estamos usando XAMMP, sí desea hacerlo utilizando XAMMP, simplemente realiza el proceso al igual que con los módulos anteriores, la URL sería: http://localhost/diplomado/html/pagina.html.

Eres libre de escoger como trabajar, el resultado será el mismo.



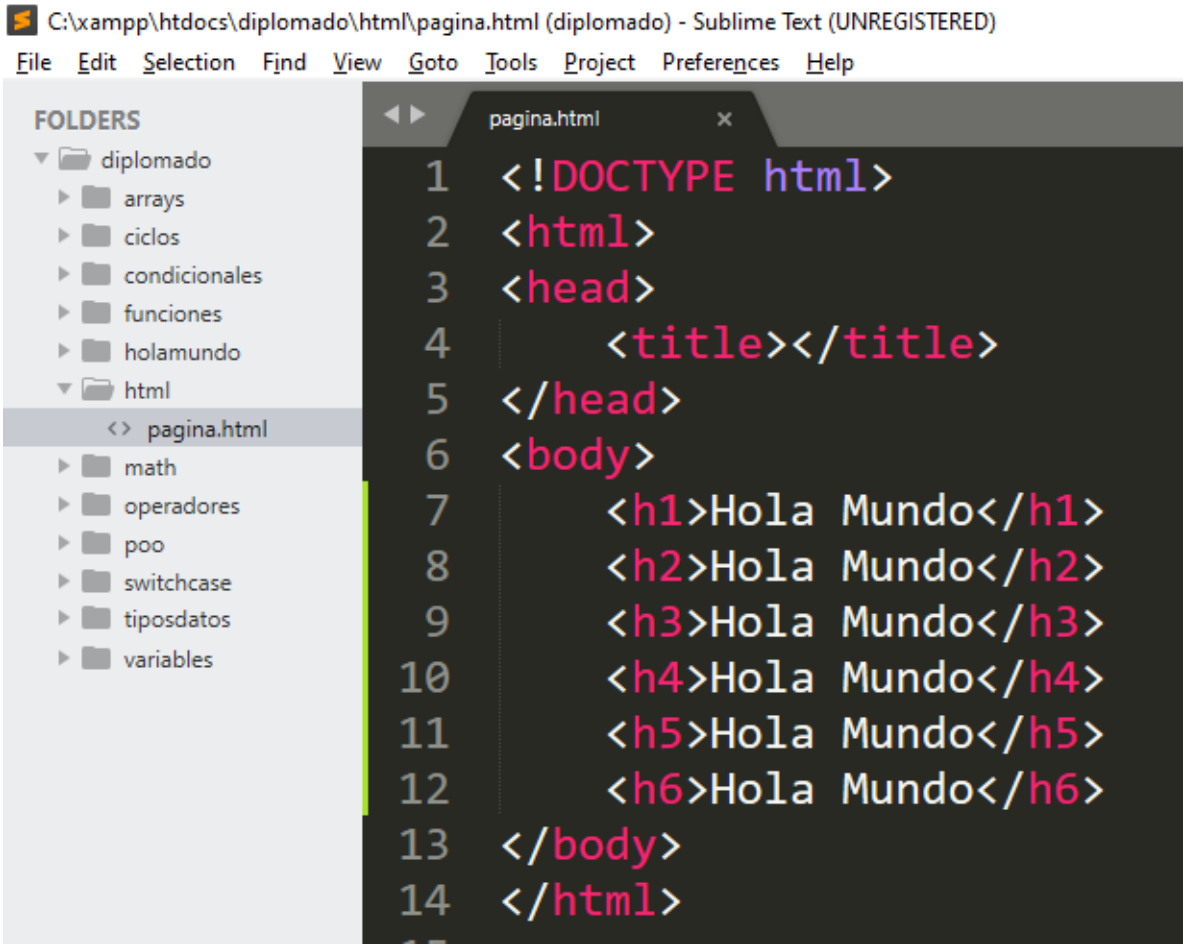
El resultado es completamente en blanco la pantalla, esto es debido a que el navegador no muestra las etiquetas de HTML. Simplemente muestra el contenido dentro de estas.

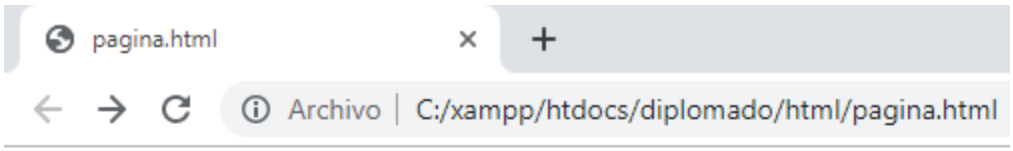
Ahora démosle contenido al nuestro documento, el "Hola Mundo". En este caso usaremos una nueva etiqueta muy particular.

TEMA 3
ENCABEZADOS

Los encabezados se definen con las etiquetas **<h1>** a **<h6>.**

<h1> define el encabezado más importante. **<h6>** define el encabezado menos importante.





Hola Mundo

Hola Mundo

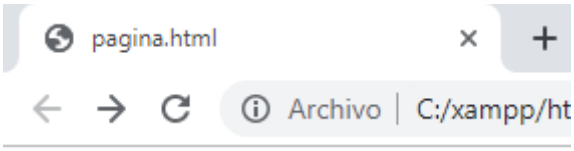
Hola Mundo

Hola Mundo

Hola Mundo

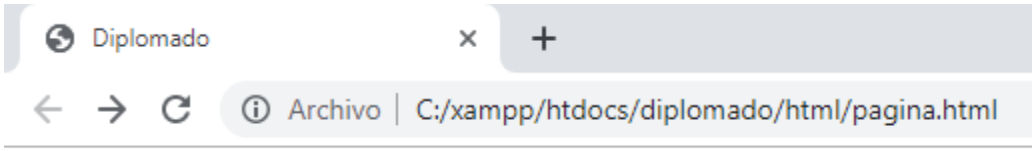
Hola Mundo

Este ha sido el resultado de nuestra primera página web. Realicemos otro cambio. HTML dentro de sus etiquetas, se encuentra: <title></title>. Title permite darle un título o nombre a nuestra página, se fijan en la parte superior aparece simplemente el nombre del archivo.



Si se realiza el cambio en la etiqueta title, el resultado sería el siguiente:

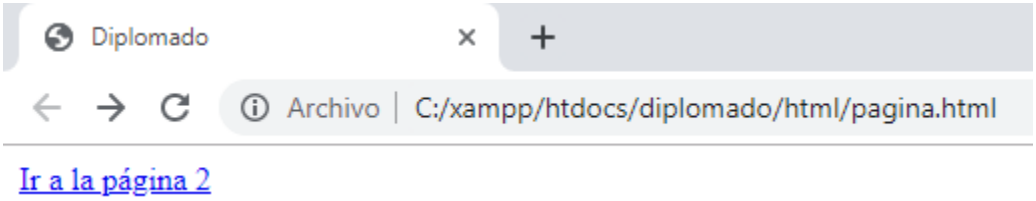
```
<head>
    <title>Diplomado</title>
</head>
```



TEMA 4
ENLACES - HIPERVINCULOS

Los enlaces permiten a los usuarios hacer clic de una página a otra y navegar en ese sentido. En HTML Los hipervínculos se definen con la <a>:





La etiqueta <a> implementa un nuevo concepto en HTML que son los atributos. Los atributos son bloques de código que se encuentran en las etiquetas de HTML.

Atributos

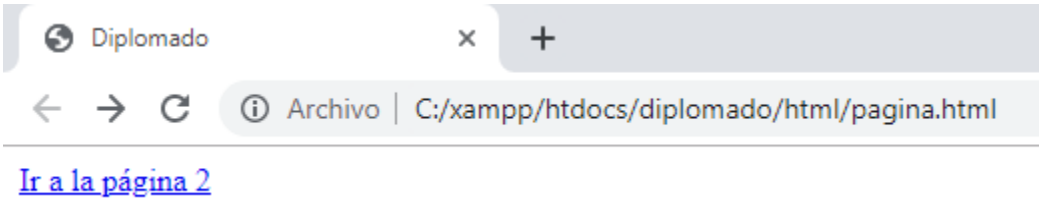
- Todos los elementos HTML pueden tener atributos
- Los atributos proporcionan información adicional sobre un elemento
- Los atributos siempre se especifican en la etiqueta de inicio

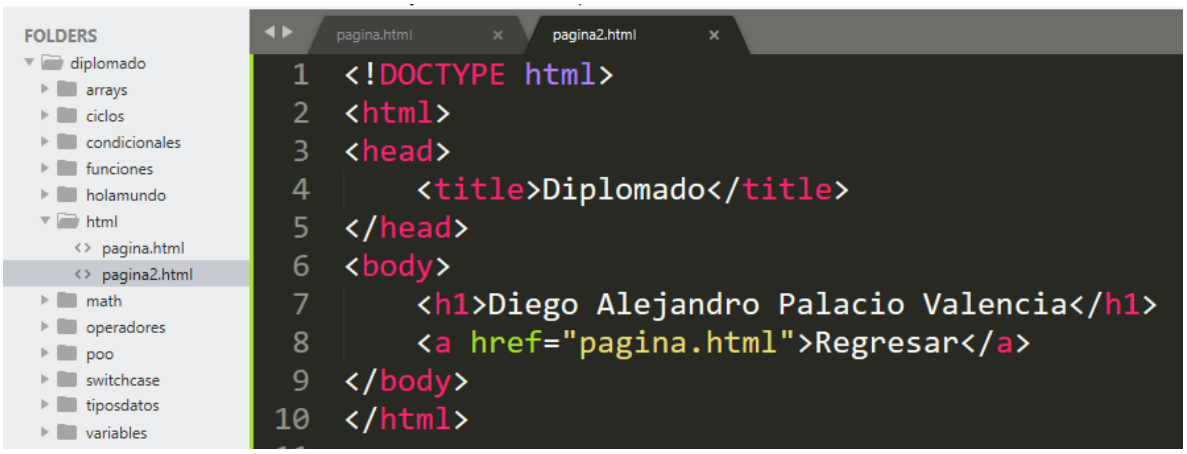
- Los atributos generalmente vienen en pares nombre / valor como: nombre = "valor"

En el caso de la etiqueta <a>, el nombre del atributo es “href” y el valor que recibe es una URL.

Creemos una segunda página, con el nombre asignado a la respectiva etiqueta: “pagina2.html” con un encabezado que diga su respectivo nombre y otro enlace para regresas. El resultado sería el siguiente:



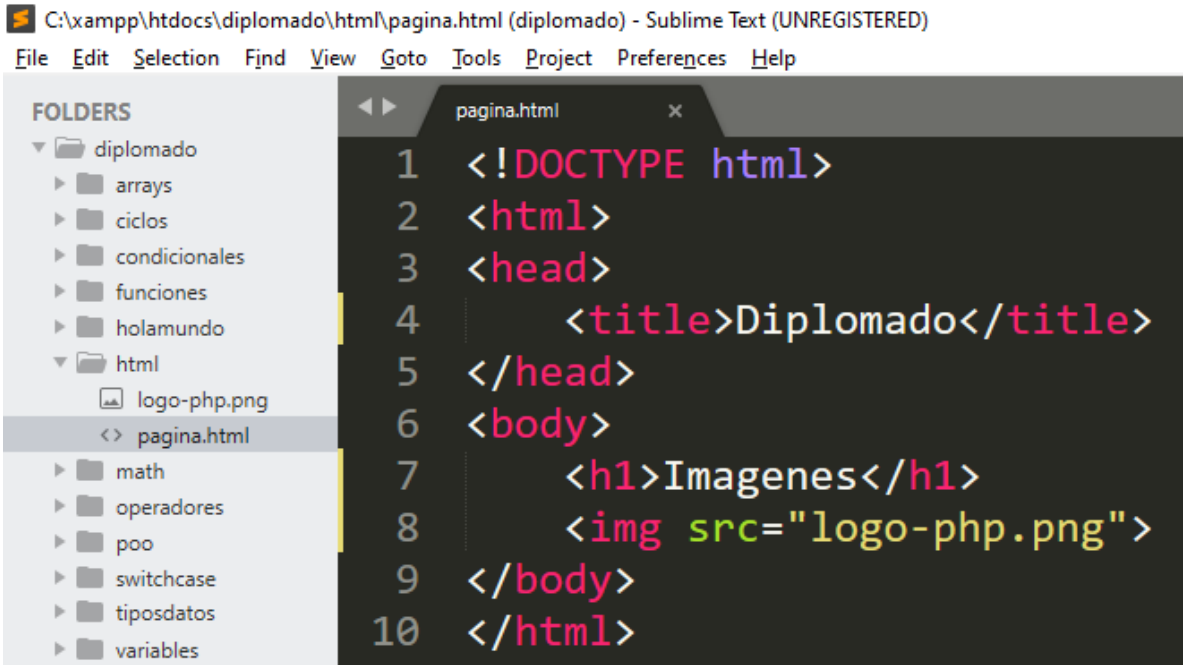



FOLDERS
diplomado
arrays
ciclos
condicionales
funciones
holamundo
html
pagina.html
pagina2.html
math
operadores
poo
switchcase
tiposdatos
variables
pagina.html
pagina2.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Diplomado</title>
5 </head>
6 <body>
7 <h1>Diego Alejandro Palacio Valencia</h1>
8 Regresar
9 </body>
10 </html>


Diplomado
Archivo | C:/xampp/htdocs/diplomado/html/pagina2.html
Diego Alejandro Palacio Valencia
Regresar

TEMA 5
IMAGENES

En HTML, las imágenes se definen con la <img>. <img> está vacía, contiene solo atributos y no tiene una etiqueta de cierre. El atributo src especifica la URL de la imagen:





Para descargar esta imagen, ve al siguiente enlace y guárdalo en folder html y del proyecto diplomado: http://politecnicodecolombia.edu.co/paquetes/php/logo-php.png.

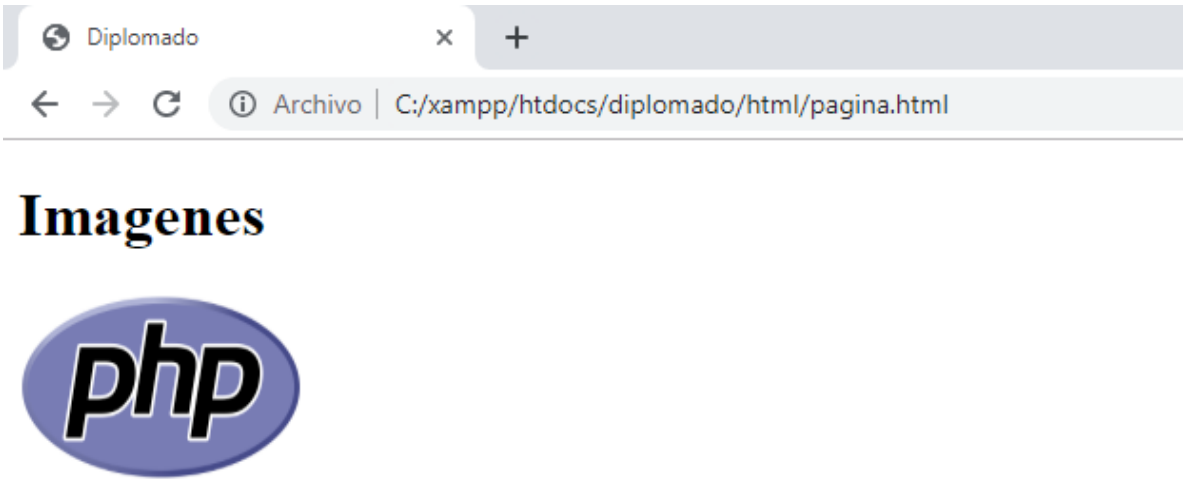
<img> contiene otros atributos muy importantes que permiten trabajar con el tamaño de las imágenes.

```
<h1>Imagenes</h1>
<img src="logo-php.png" width="" height="">
```

- Width: Define el ancho de la imagen.
- Heigth: Define el alto de la image.

Los valores que reciben estos atributos son, pixeles o porcentajes, en este caso manejemos los valores con pixeles. Veamos.

```
<h1>Imagenes</h1>
<img src="logo-php.png" width="150" height="100">
```



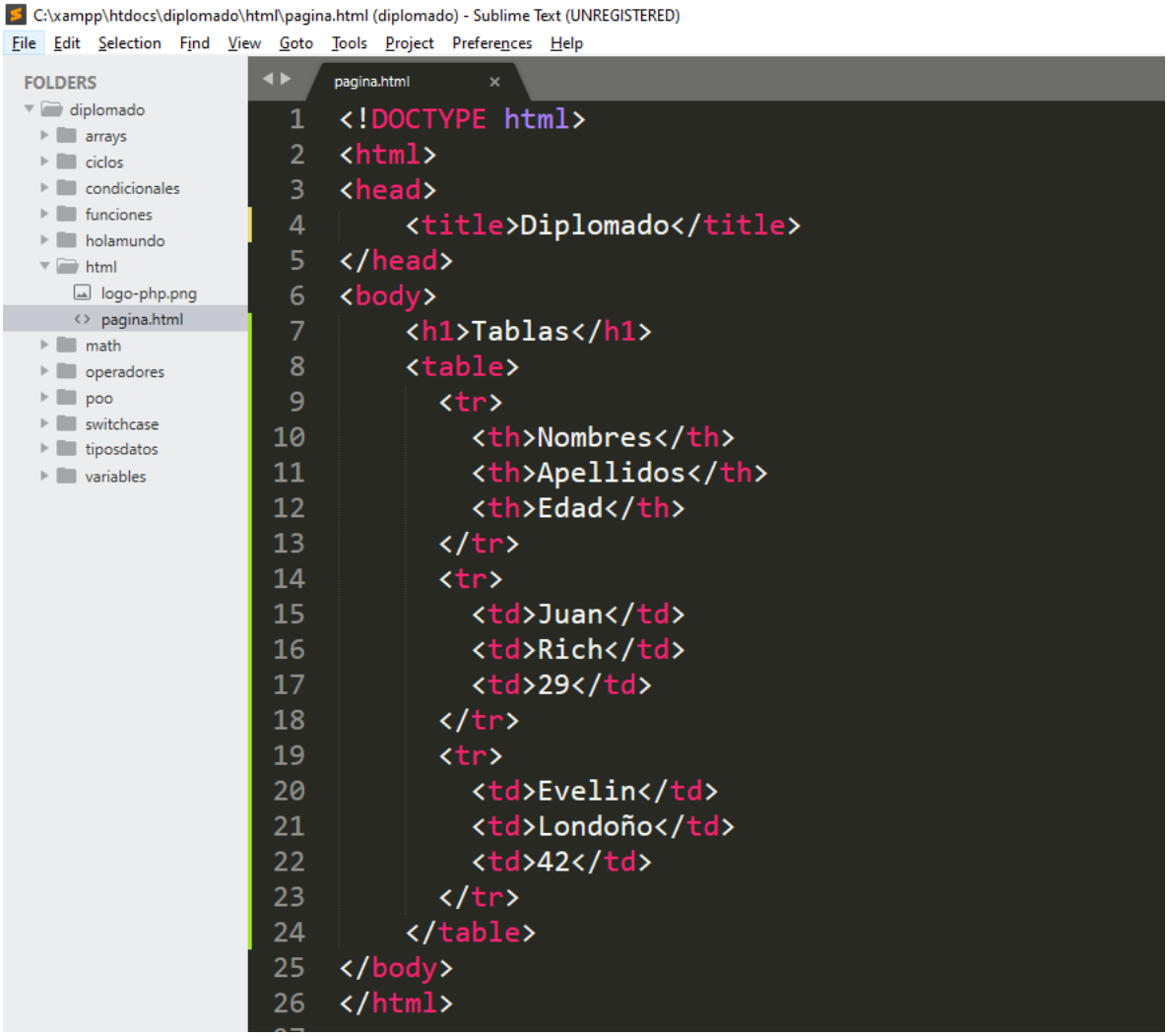
Jugando un poco con las etiquetas podemos utilizar enlaces e imágenes en un solo bloque de código, digamos que deseamos que cuando demos click en la imagen del logo de PHP vayamos a la página oficial de PHP. Este sería el código:

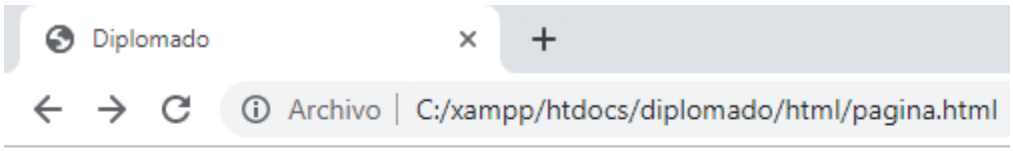


TEMA 6
TABLAS

Una tabla HTML se define con la etiqueta <table> </tabla>.

Cada fila de la tabla se define con la etiqueta **<tr><tr>**. Un encabezado de tabla se define con la etiqueta **<th></th>.** Por defecto, los encabezados de las tablas están en negrita y centrados. Una tabla de datos / celda se define con la etiqueta **<td></td>.**





Tablas

Nombres	Apellidos	Edad
Juan	Rich	29
Evelin	Londoño	42

A simple vista parece una tabla, hagamos uso de una de los atributos de las tablas, en este caso: "boder".

```
<body>
    <h1>Tablas</h1>
    <table border="1">
      <tr>
        <th>Nombres</th>
        <th>Apellidos</th>
        <th>Edad</th>
```



Tablas

Nombres	Apellidos	Edad
Juan	Rich	29
Evelin	Londoño	42

Cambia un poco más visualmente. La estructura de una tabla la podemos definir en tres partes.

- El encabezado.
- La fila
- La columna.

El encabezado hace referencia directamente a los campos que representará la tabla y las etiquetas que lo definen son: <tr> y <th>, de la siguiente forma:

```
<tr>
  <th>Nombres</th>
  <th>Apellidos</th>
  <th>Edad</th>
</tr>
```

Las filas hacen referencia a cada uno de registros que representa la tabla frente a sus encabezados, cada fila representa un registro en la tabla y las etiquetas son: <tr>.

```
<tr>
  <td>Juan</td>
  <td>Rich</td>
  <td>29</td>
</tr>
```

Las columnas son datos específicos dentro de una fila en relación con un encabezado de la tabla, la etiqueta que la representa es: <td>

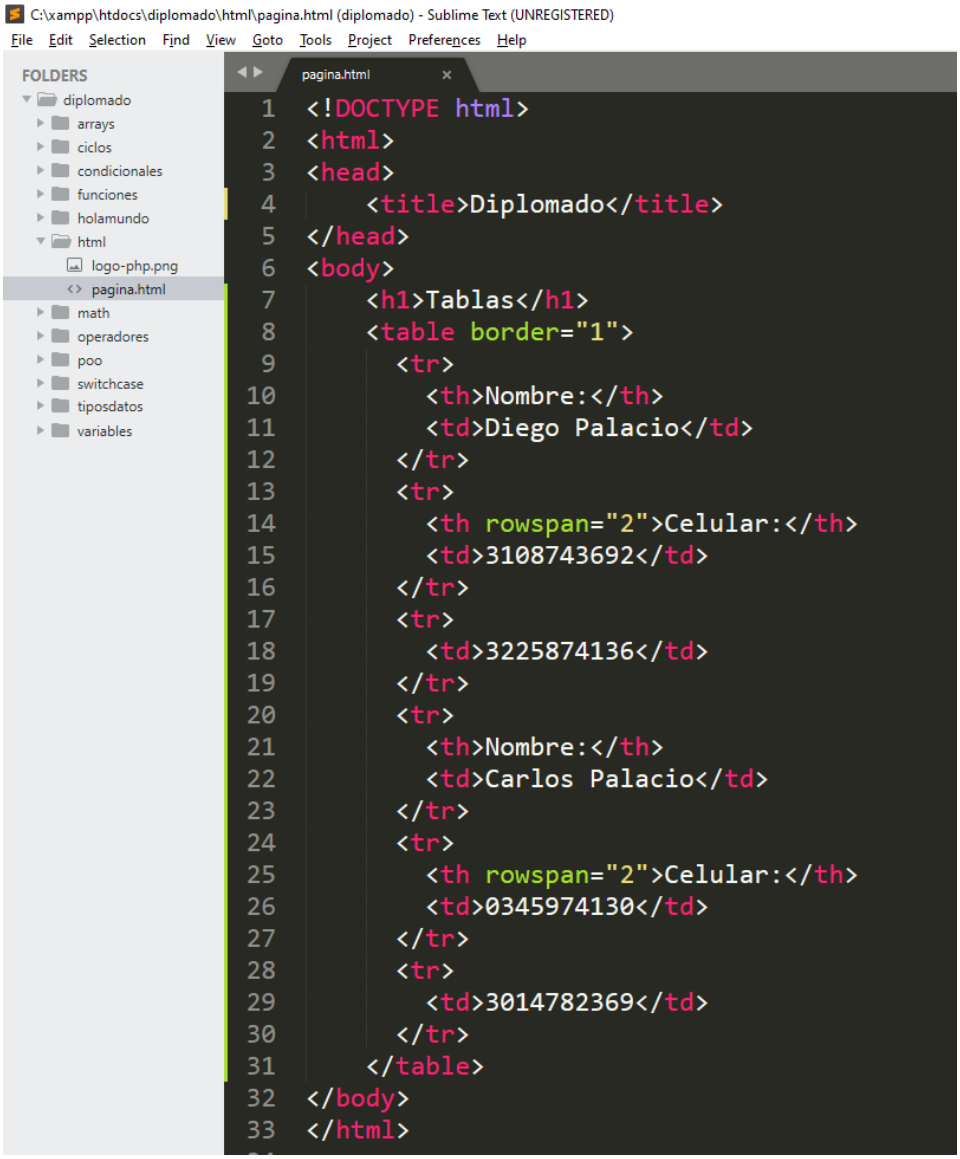
```
<tr>
  <td>Juan</td>
  <td>Rich</td>
  <td>29</td>
</tr>
```

Existen dos atributos muy importantes cuando se trabaja con filas, permite cambiar el estilo, orden y tamaño de la misma, estos son:

- Rowspan
- Colspan

Rowspan

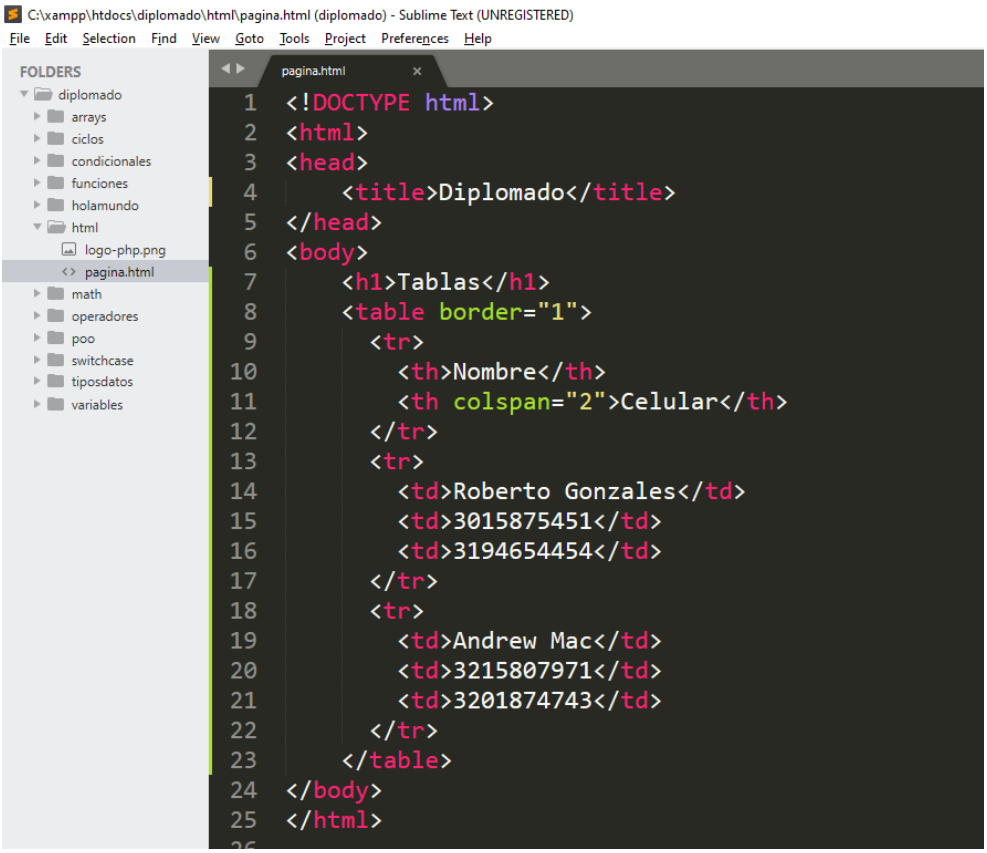
Rowspan permite que una celda ocupe más de una fila.

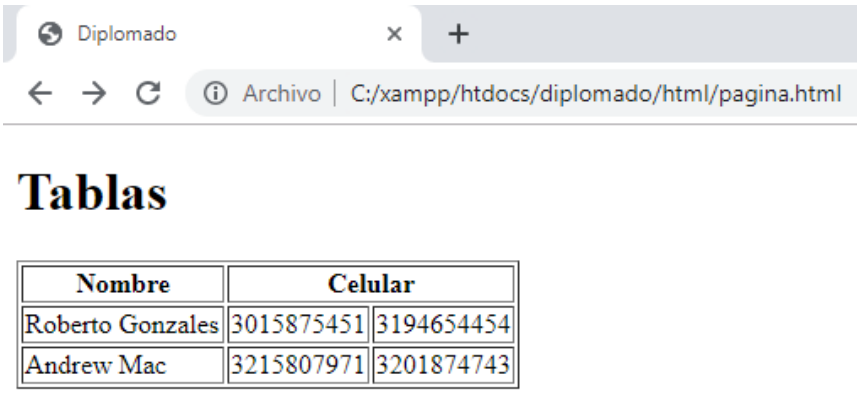




Colspan

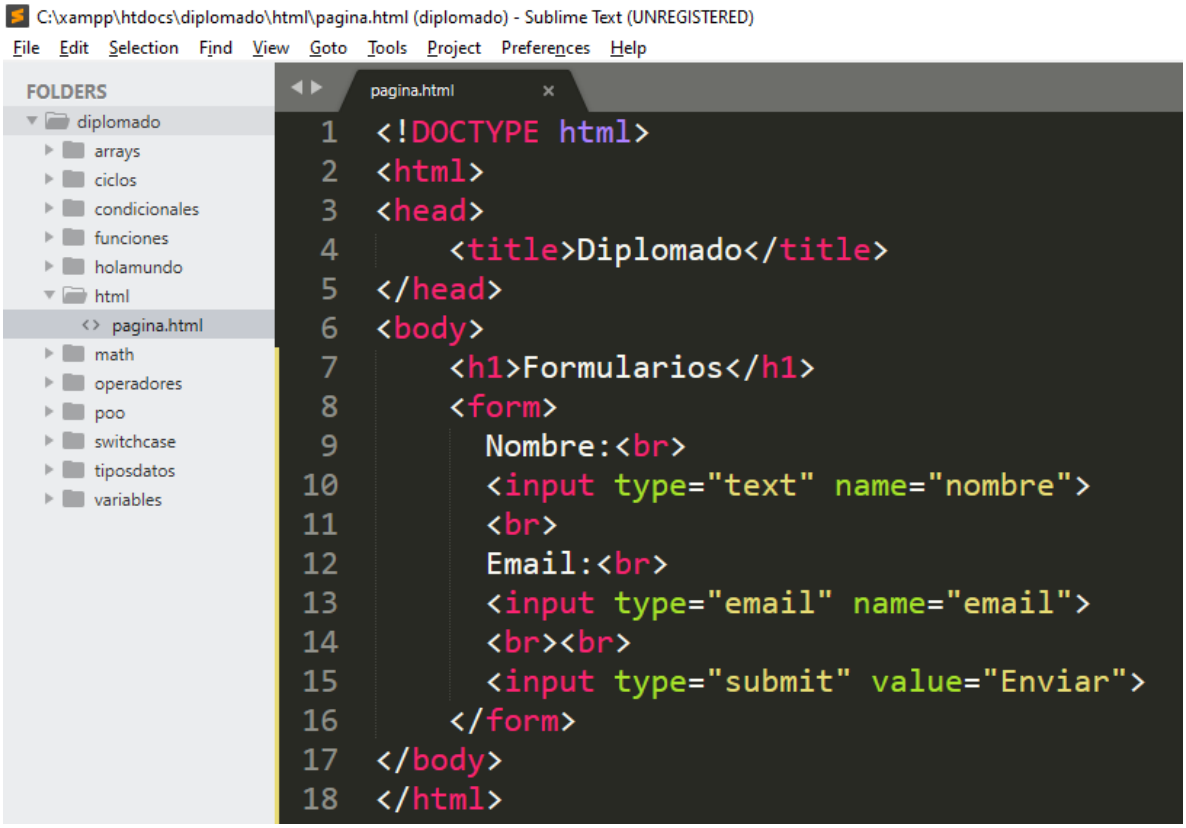
Colspan permite que una celda ocupe más de una columna.

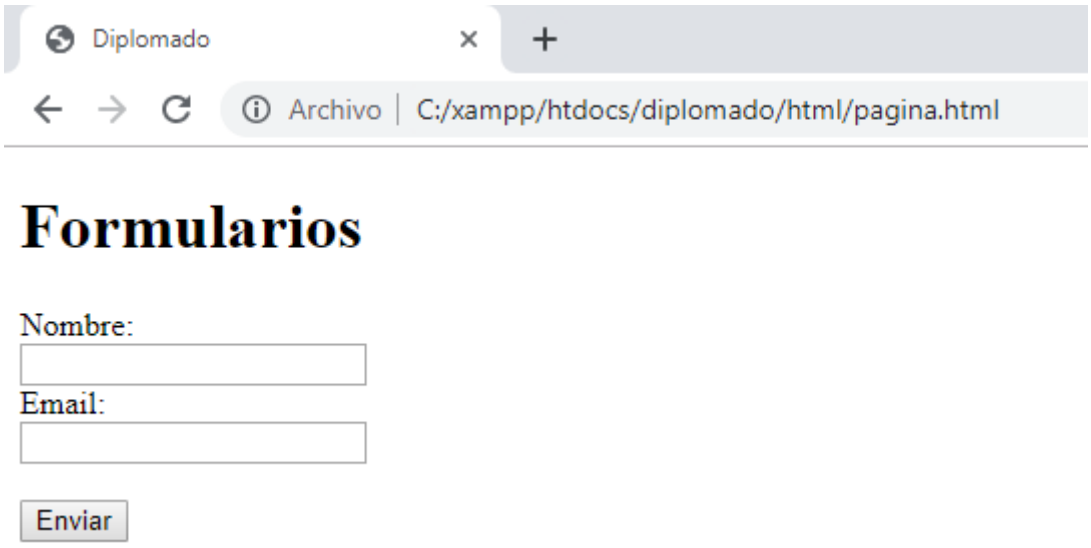



Diplomado
Archivo | C:/xampp/htdocs/diplomado/html/pagina.html
Tablas
Nombre
Celular
Roberto Gonzales
3015875451
3194654454
Andrew Mac
3215807971
3201874743

TEMA 7
FORMULARIOS

El **<form></form>** HTML define un formulario que se utiliza para recopilar la entrada del usuario.





Un formulario HTML contiene elementos de formulario.

Los elementos de formulario son diferentes tipos de elementos de entrada de datos, como campos de texto, casillas de verificación, botones de opción, botones de envío y más, pero todos deben de ir dentro de las etiquetas de **<form></form>.**

Los elementos de entrada son:

- Input
- Select
- Texarea
- Button

El elemento <input>

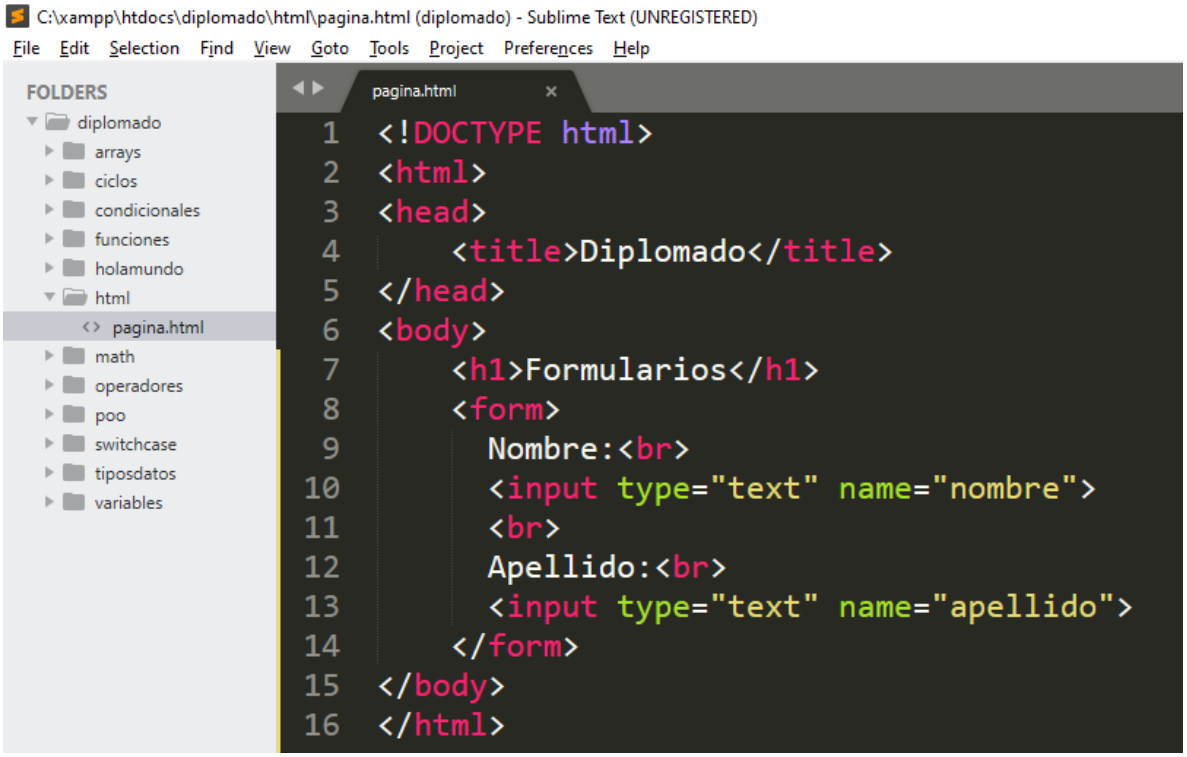
El elemento **<input>** es el elemento de forma más importante. El elemento **<input>** se puede mostrar de varias maneras, dependiendo del atributo de tipo.

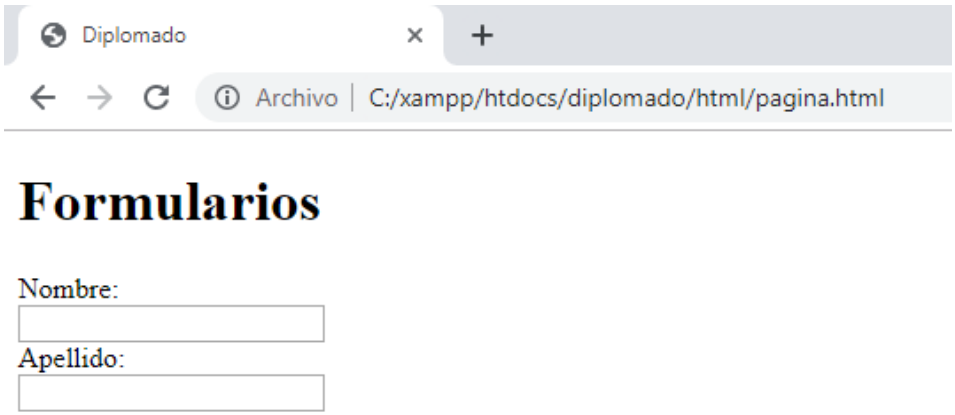
Aquí hay unos ejemplos:

Tipo	Descripción
<input type = "text">	Define un campo de entrada de texto de una línea.
<input type = "radio">	Define un botón de radio (para seleccionar una de muchas opciones)
<input type = "submit">	Define un botón de envío (para enviar el formulario)

Entrada de texto

<input type="text"> define un campo de entrada de una línea para la entrada de texto:




Diplomado
Archivo | C:/xampp/htdocs/diplomado/html/pagina.html
Formularios
Nombre:
Apellido:

Atributo type <input>

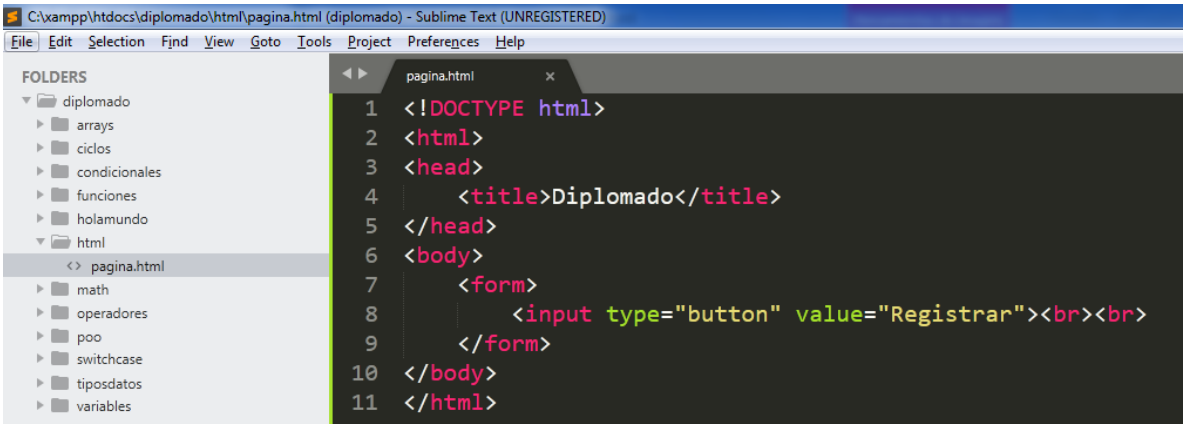
El atributo de la etiqueta **<input>** es sumamente funcional por todas las características técnicas en las que se puede usar en relación con los formularios. Esta se puede especificar para todo tipo de datos, desde que solo reciba números hasta una fecha, incluso importar imágenes, estos son algunos tipos:

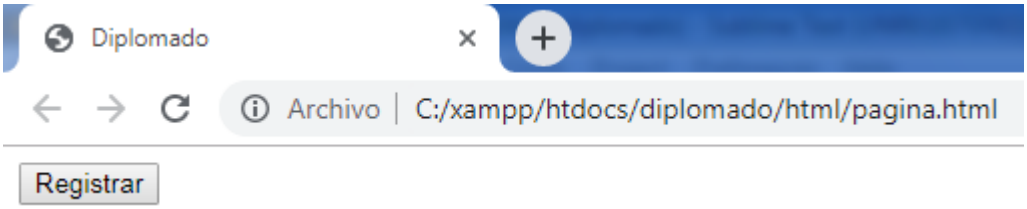
Tipo
<input type="button">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="image">
<input type="month">
<input type="number">
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="text">
<input type="time">
<input type="url">
<input type="week">

Ahora vemos algunos de los principales types en código y la forma en que operan en los formularios.

Type Button

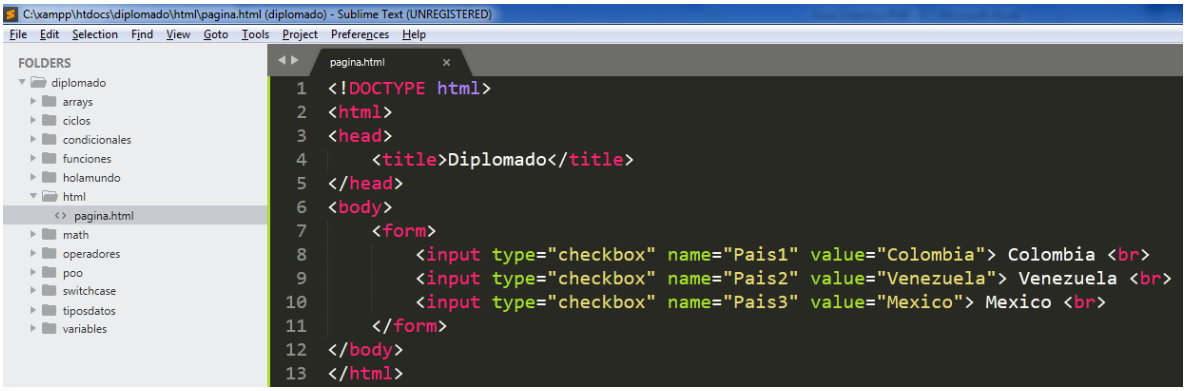
<inputtype="button"> Define un botón:

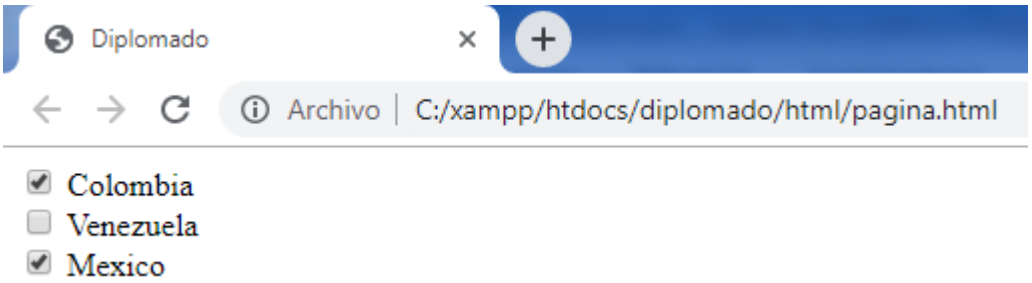




Type Checkbox

<input type="checkbox"> define una **casilla de verificación**. Las casillas de verificación le permiten al usuario seleccionar CERO o MÁS opciones de un número limitado de opciones.

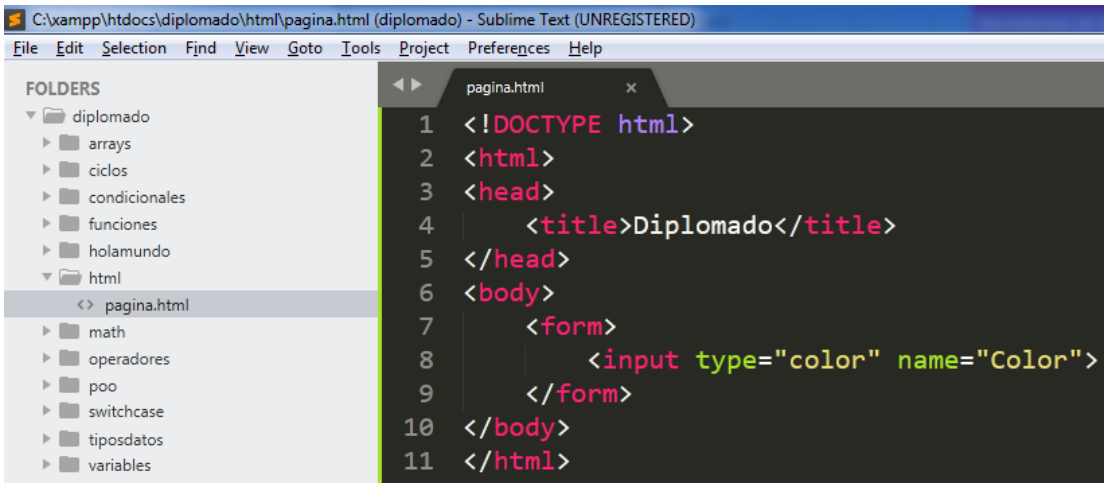


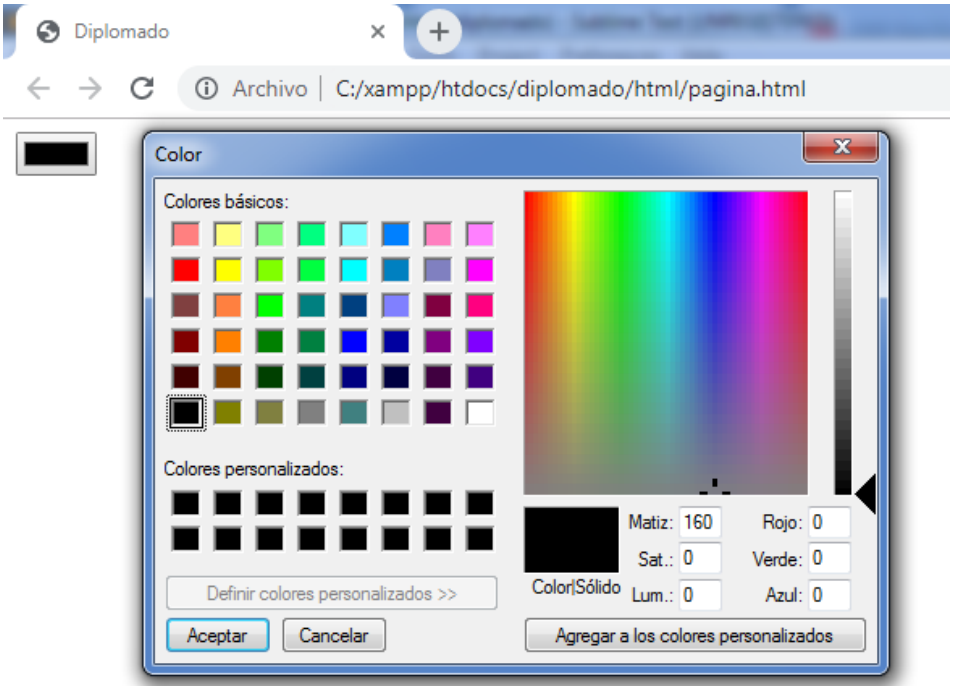


Type Color

<input type="color"> utiliza para campos de entrada que deben contener un color.

Dependiendo de la compatibilidad del navegador, puede aparecer un selector de color en el campo de entrada.

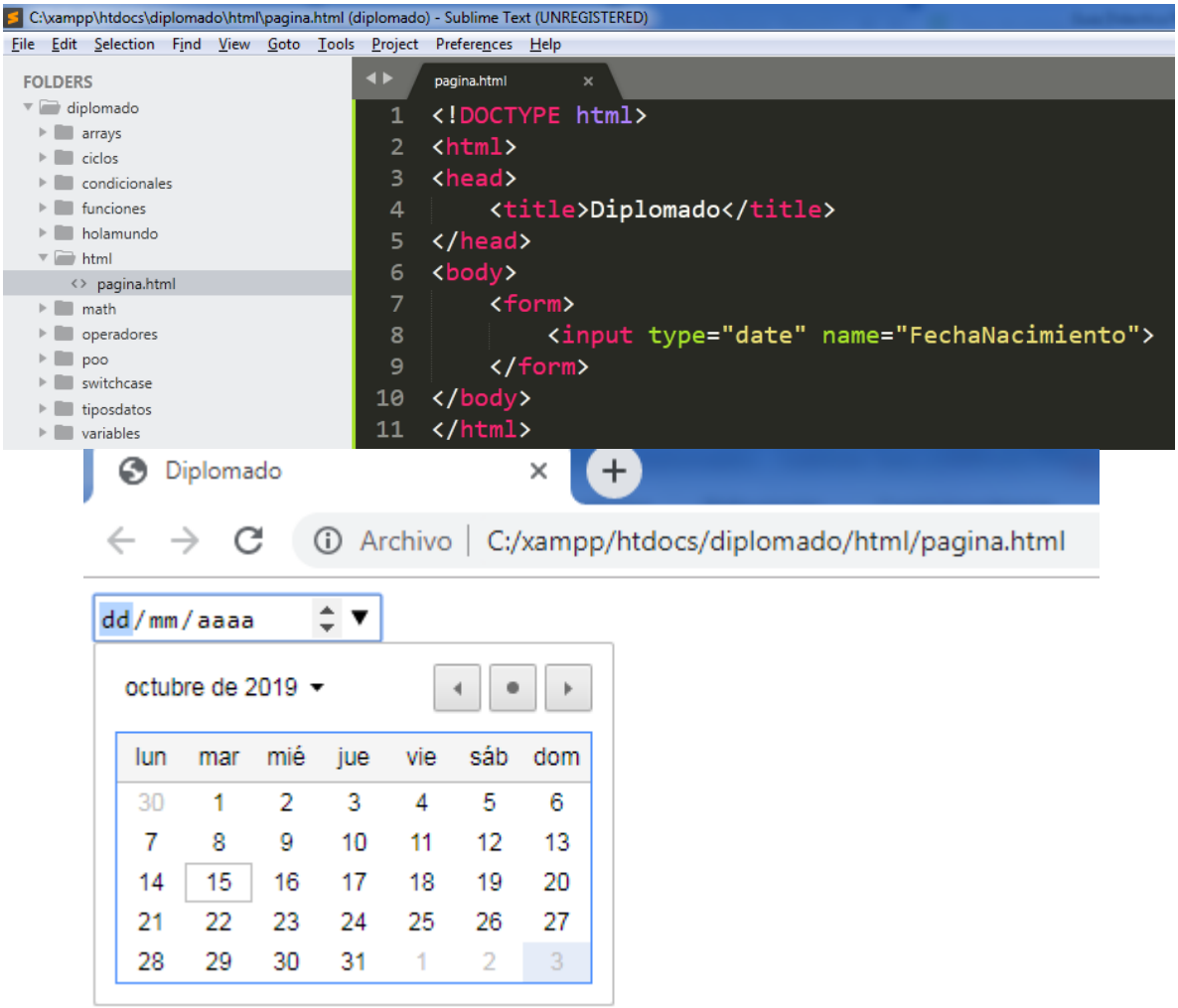




Type Date

<input type="date"> se utiliza para los campos de entrada que deben contener una fecha.

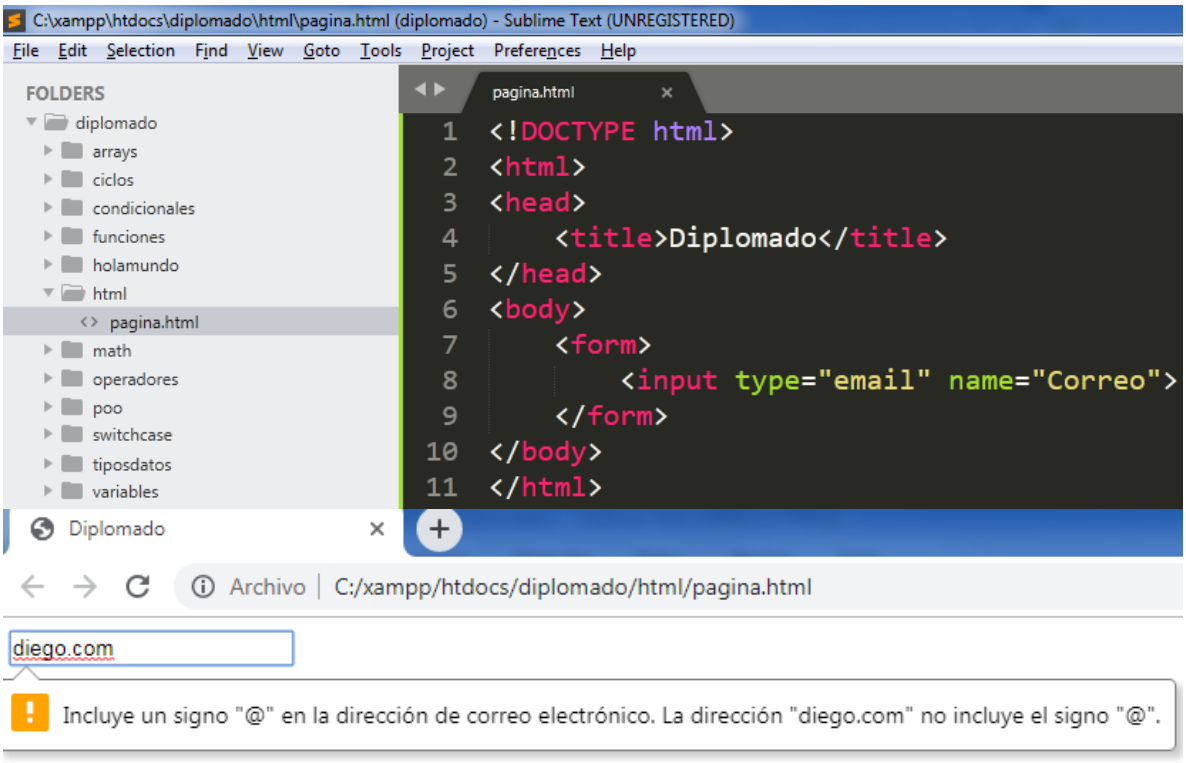
Dependiendo del soporte del navegador, un selector de fecha puede aparecer en el campo de entrada.



Type Email

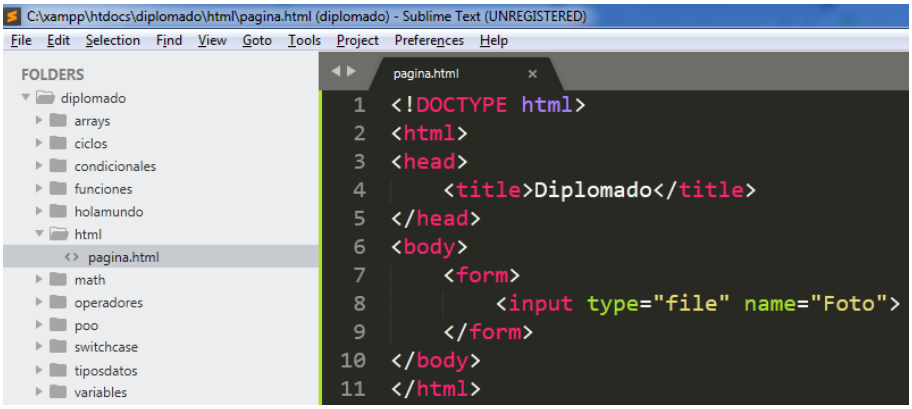
<input type="email"> se utiliza para campos de entrada que deben contener una dirección de correo electrónico.

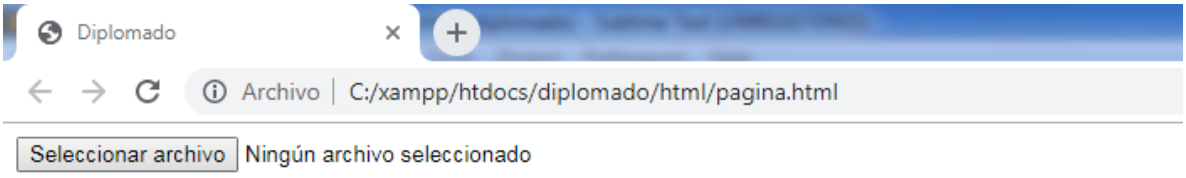
Dependiendo del soporte del navegador, la dirección de correo electrónico se puede validar automáticamente cuando se envía.

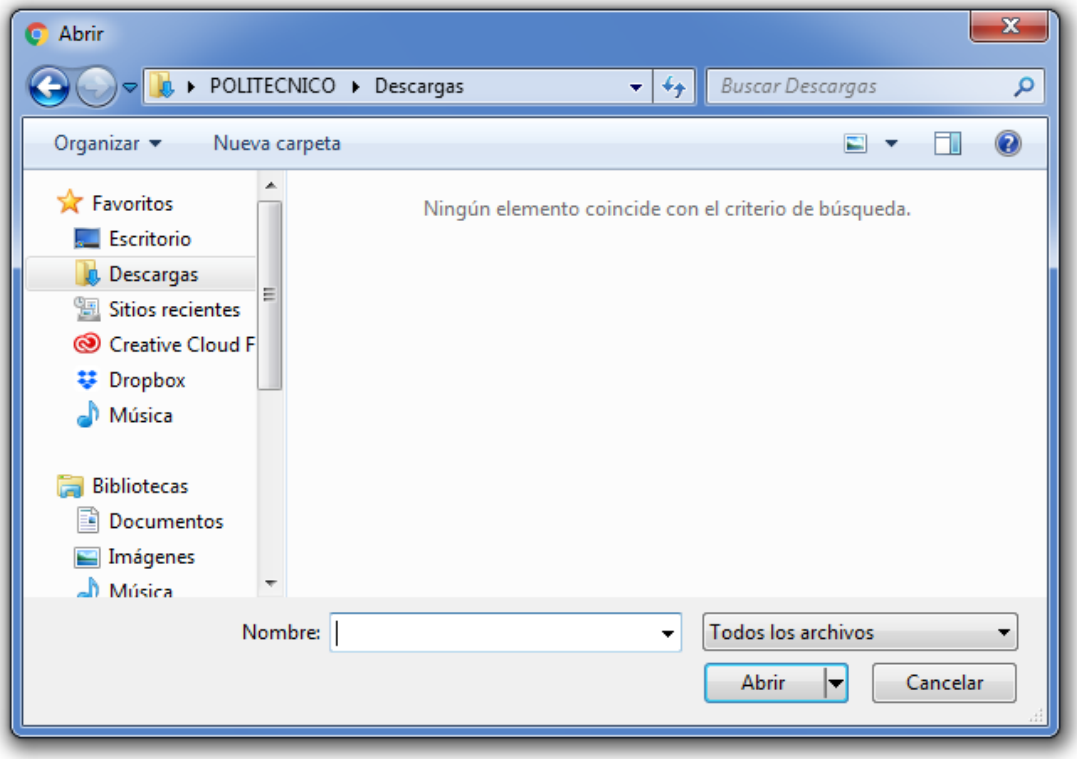


Type File

<input type="file"> define un campo de selección de archivo y un botón "Examinar" para la carga de archivos.

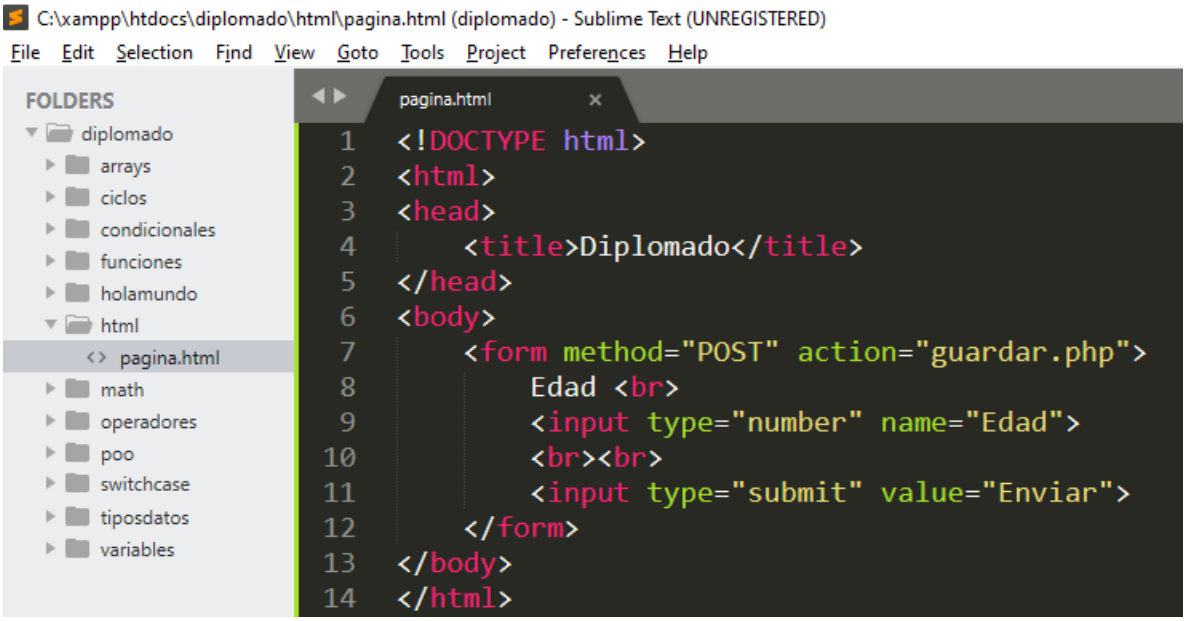


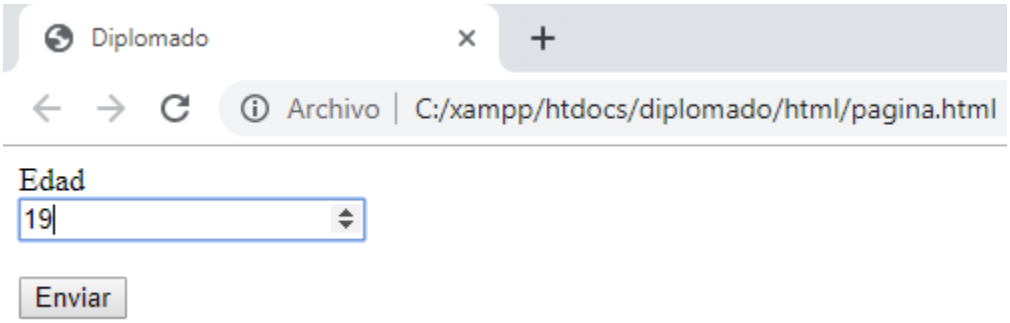




Type Number

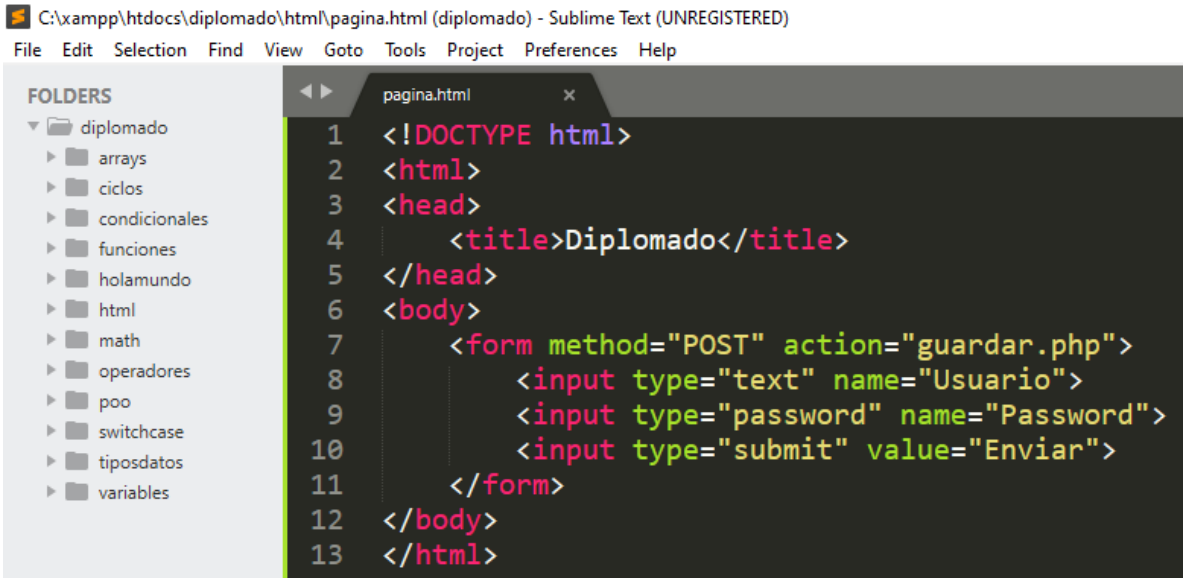
<input type="number"> define un campo de entrada numérico.

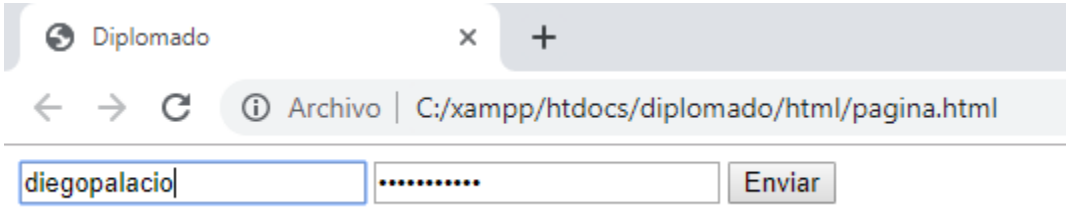




Type Password

<input type="password"> define un campo de contraseña, con la particularidad de que los caracteres se ven representados en asteriscos:



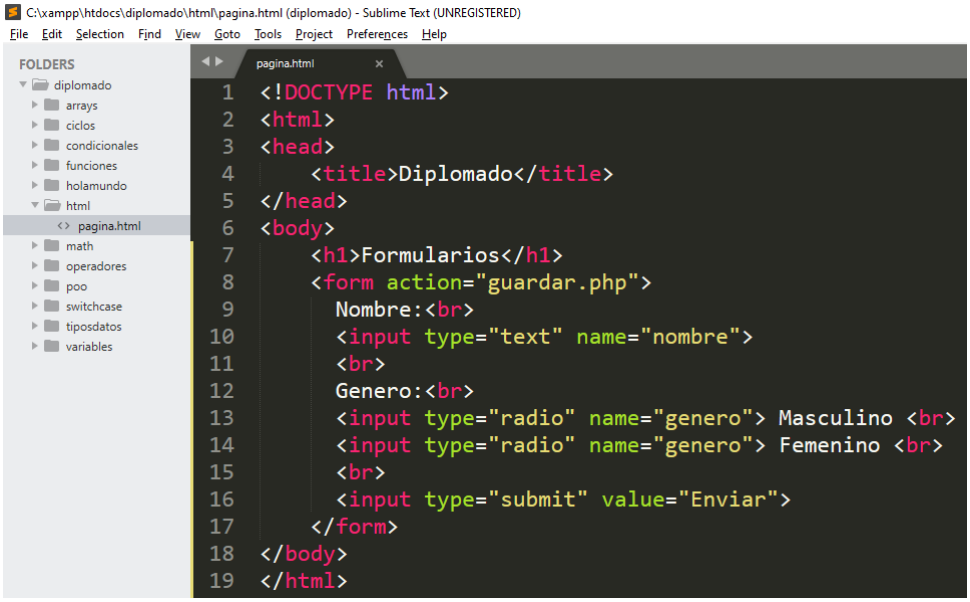


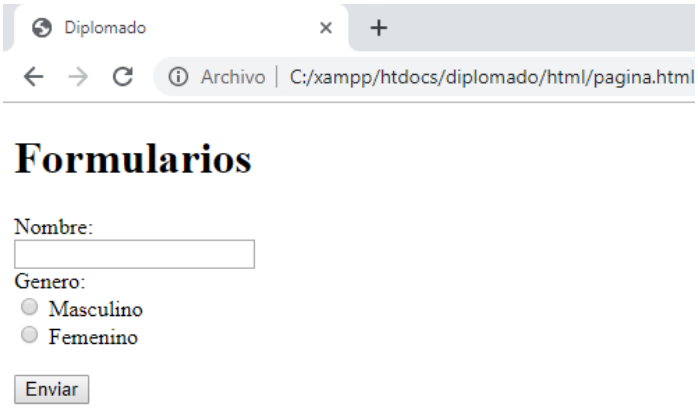
Type Submit

<input type="submit"> define un botón para enviar los datos del formulario a un controlador de formulario.

El controlador de formulario suele ser una página de servidor con un script para procesar datos de entrada (En este caso PHP, pero más adelante veremos su uso).

El controlador de formulario se especifica en el atributo de acción del formulario:





Atributo Action

El **action** atributo define la acción a realizar cuando se envía el formulario.

Normalmente, los datos del formulario se envían a una página web en el servidor cuando el usuario hace clic en el botón Enviar.

En el ejemplo anterior, los datos del formulario se envían a una página en el servidor llamada "guardar.php". Esta página contiene un script del lado del servidor que maneja los datos del formulario:

```
<form action="guardar.php">
```

Nota: Si action se omite el atributo, la acción se establece en la página actual.

Atributo Method

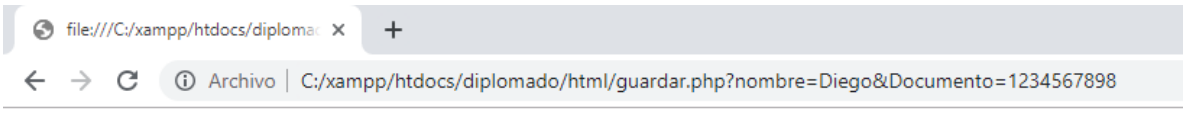
El atributo method específica el método HTTP (GET o POST) que se utilizará al enviar los datos del formulario:

Method GET

```
<form action="guardar.php" method="GET">
```

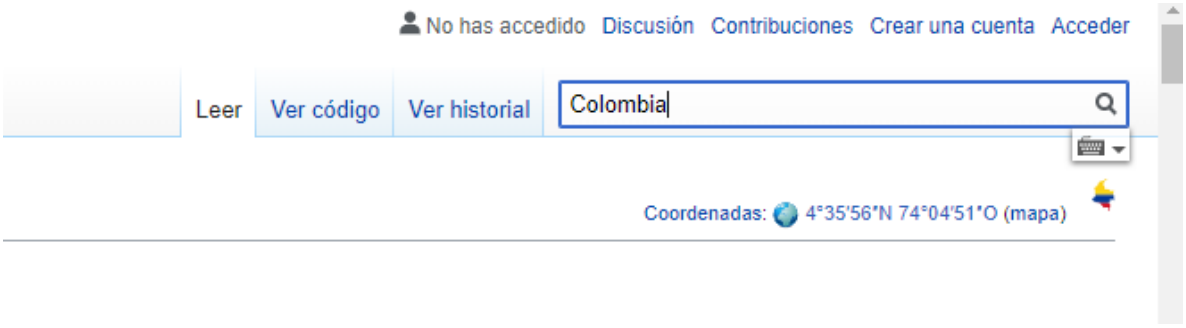
El método predeterminado al enviar datos del formulario es GET.

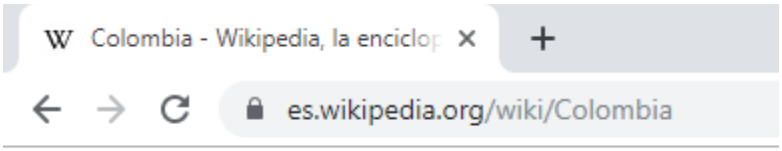
Sin embargo, cuando se utiliza GET, los datos del formulario enviado serán visibles en el campo de dirección de la página:



Notas sobre GET:

- Agrega datos de formulario a la URL en pares de nombre / valor
- La longitud de una URL es limitada (aproximadamente 3000 caracteres)
- ¡Nunca use GET para enviar datos confidenciales! (será visible en la URL)
- Útil para envíos de formularios donde un usuario desea marcar el resultado
- GET es mejor para datos no seguros, como cadenas de consulta en Wikipedia





Method POST

```
<form action="guardar.php" method="POST">
```

Utilice siempre POST si los datos del formulario contienen información confidencial o personal. El método POST no muestra los datos del formulario enviado en el campo de dirección de página.

Notas sobre POST:

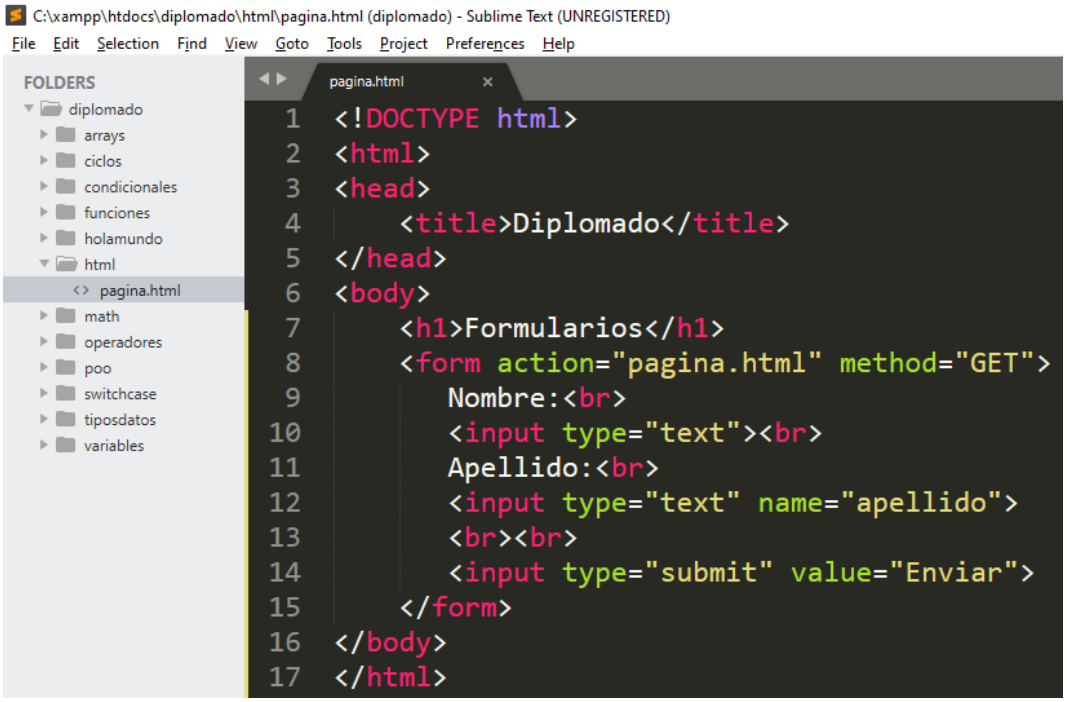
- POST no tiene limitaciones de tamaño y puede usarse para enviar grandes cantidades de datos.
- Los envíos de formularios con POST no pueden marcarse.

Atributo Name

Cada campo de entrada debe tener un atributo name para ser enviado.

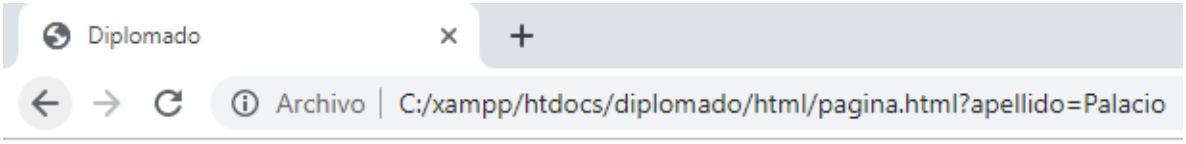
Si name se omite el atributo, los datos de ese campo de entrada no se enviarán en absoluto.

Este ejemplo solo enviará el campo de entrada "Apellido". Dado que nombre no cuenta con el atributo name.


C:\xampp\htdocs\diplomado\html\pagina.html (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
diplomado
arrays
ciclos
condicionales
funciones
holamundo
html
pagina.html
math
operadores
poo
switchcase
tiposdatos
variables
pagina.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Diplomado</title>
5 </head>
6 <body>
7 <h1>Formularios</h1>
8 <form action="pagina.html" method="GET">
9 Nombre:
10 <input type="text">
11 Apellido:
12 <input type="text" name="apellido">
13

14 <input type="submit" value="Enviar">
15 </form>
16 </body>
17 </html>


Diplomado
file:///C:/xampp/htdocs/diplomado/html/pagina.html
Formularios
Nombre:
Apellido:
Palacio
Enviar


Diplomado
Archivo | C:/xampp/htdocs/diplomado/html/pagina.html?apellido=Palacio

Entrada de Selección de Lista Desplegable

El **<select> </select>** elemento define una lista desplegable:



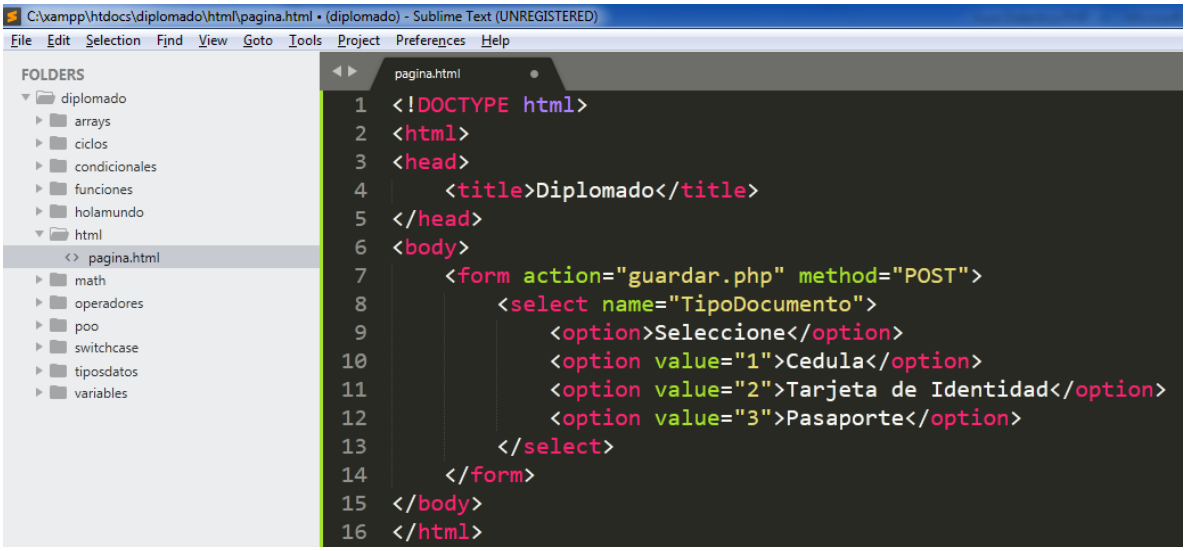


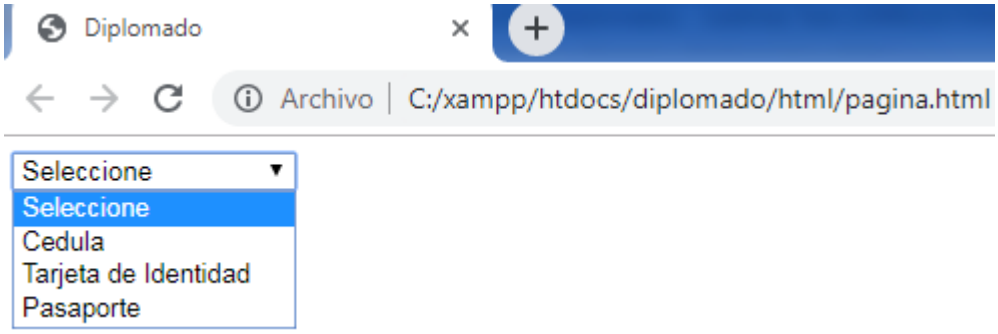
Dentro de la etiqueta **<option></option>** van los valores a seleccionar dentro de la lista desplegable. En ese orden de ideas, el atributo **value** contiene el mismo valor u otro que se desea almacenar.

Por ejemplo, representemos la siguiente información en una lista desplegable:

Código	Valor
1	Cédula
2	Tarjeta de Identidad
3	Pasaporte

El código estará contenido en el atributo **value** y el valor en la etiqueta **<option></option>**, de la siguiente forma.

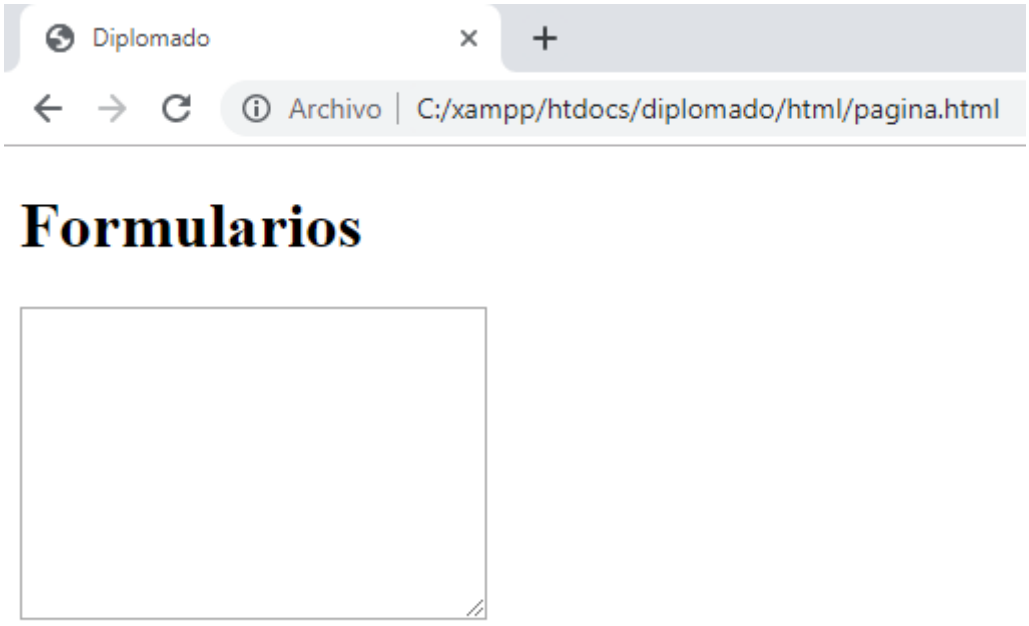




Textarea

<textarea> define un campo de entrada de varias líneas (un área de texto):





Rows

El atributo rows permite definir el número de filas del campo de texto.

Cols

El atributo cols permite definir el número de columnas del campo de texto.

Las etiquetas de entradas cuentan algunos atributos y restricciones muy especiales. Aquí la lista de algunas y su uso.

Atributo	Descripción
checked	Especifica que se debe preseleccionar un campo de entrada cuando se carga la página (para type = "checkbox" o type = "radio")
disabled	Especifica que un campo de entrada debe estar deshabilitado
max	Especifica el valor máximo para un campo de entrada
maxlength	Especifica el número máximo de caracteres para un campo de entrada
min	Especifica el valor mínimo para un campo de entrada
pattern	Especifica una expresión regular para verificar el valor de entrada
readonly	Especifica que un campo de entrada es de solo lectura (no se puede cambiar)
required	Especifica que se requiere un campo de entrada (debe completarse)
size	Especifica el ancho (en caracteres) de un campo de entrada
step	Especifica los intervalos de números legales para un campo de entrada
value	Especifica el valor predeterminado para un campo de entrada

Checked

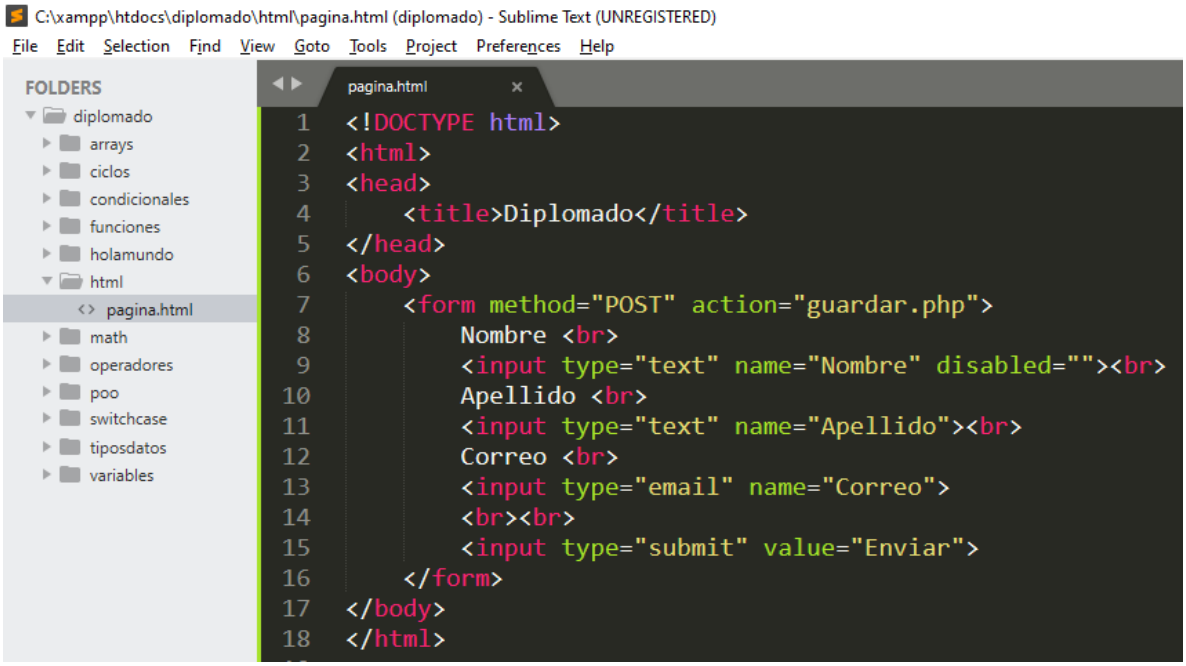
Preselecciona un checkbox automáticamente desde que la página es abierta.

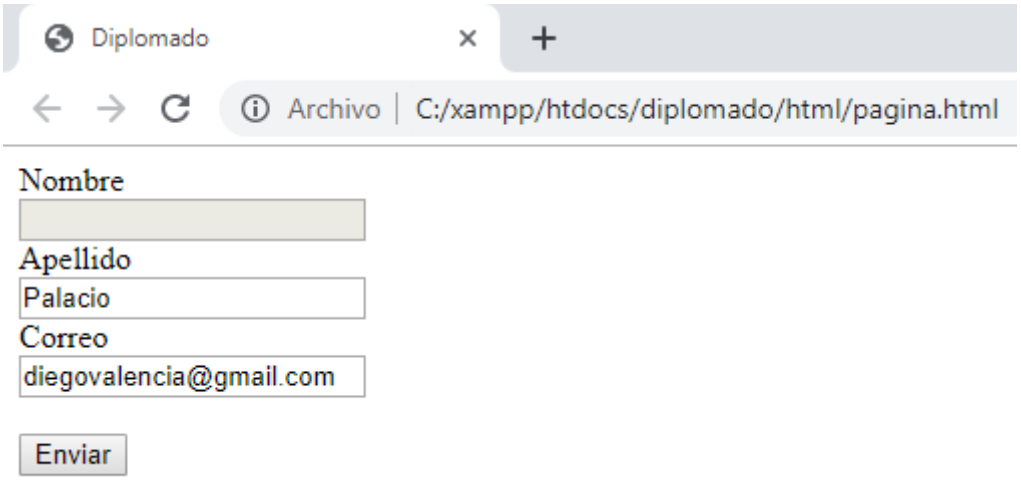




Disable

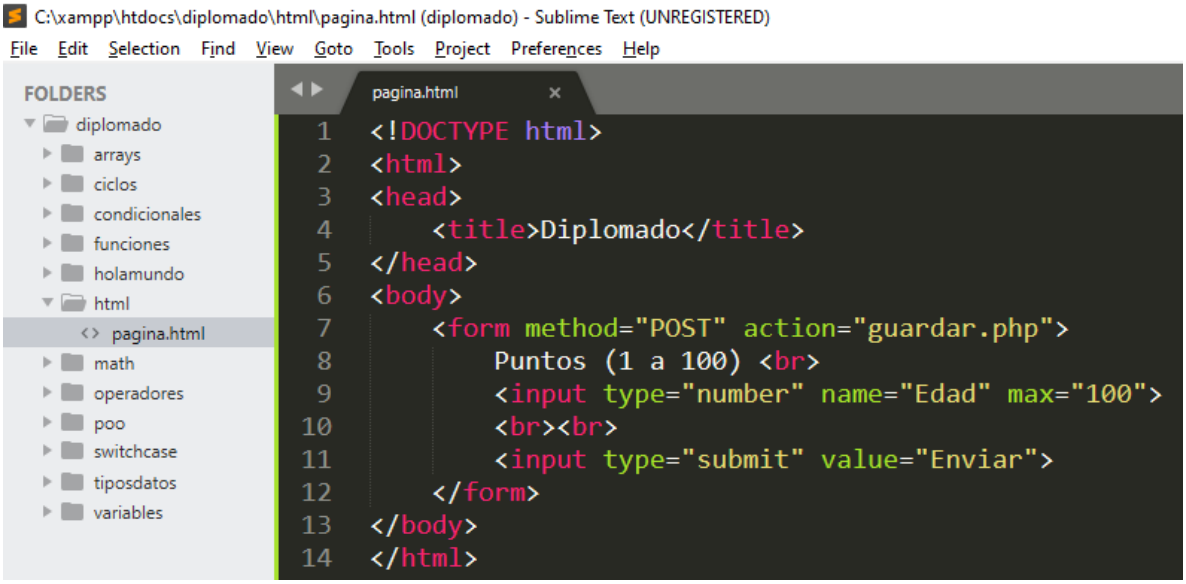
Deshabilita uno o varios campos de entrada.

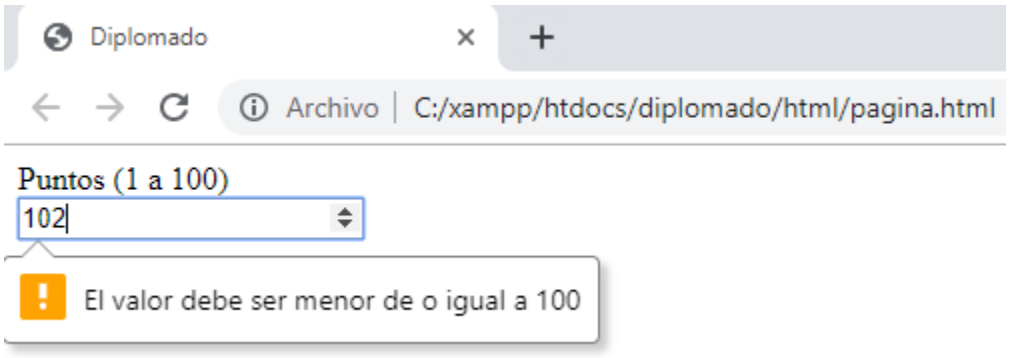




Max

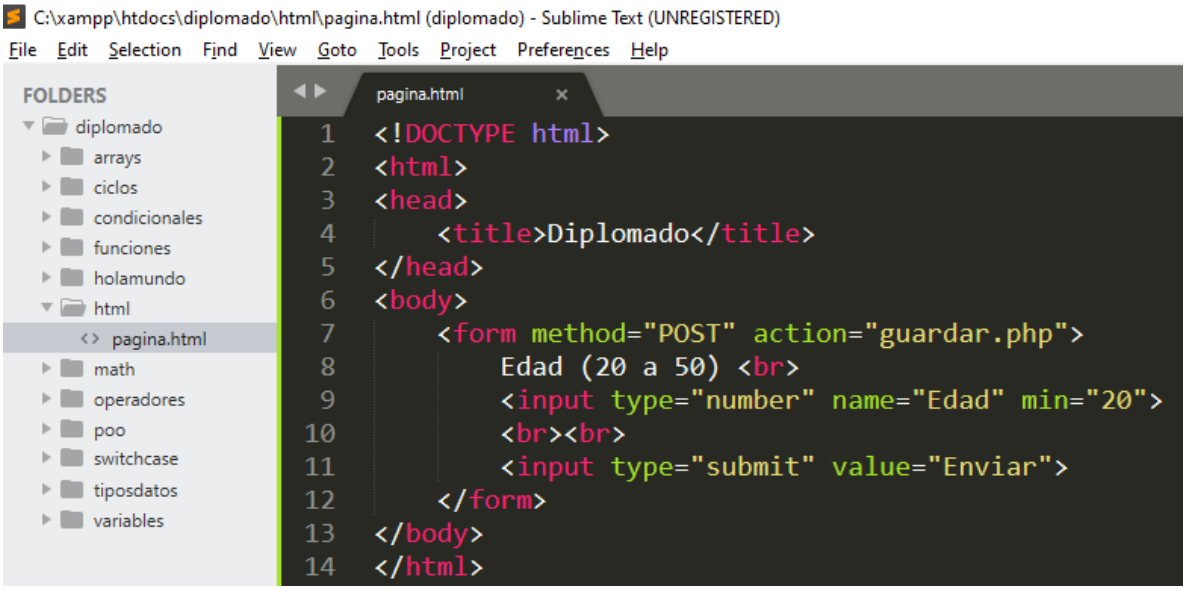
Especifica el valor máximo para un campo de entrada.

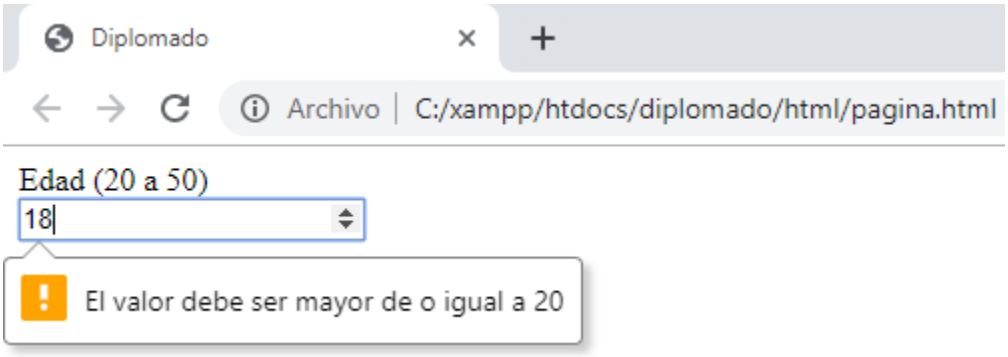




Min

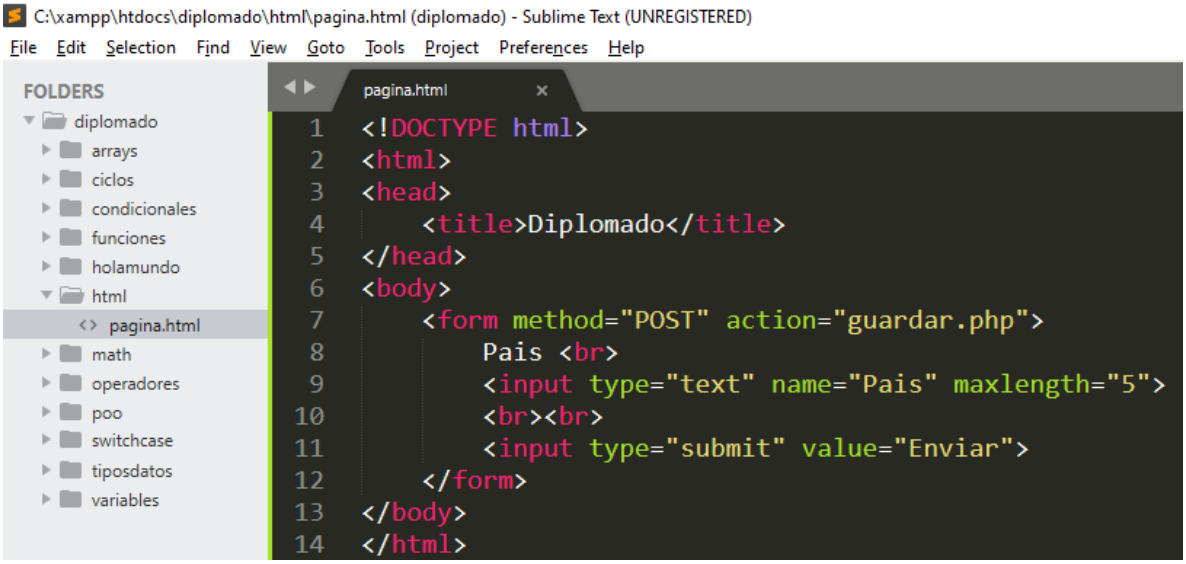
Especifica el valor mínimo para un campo de entrada.

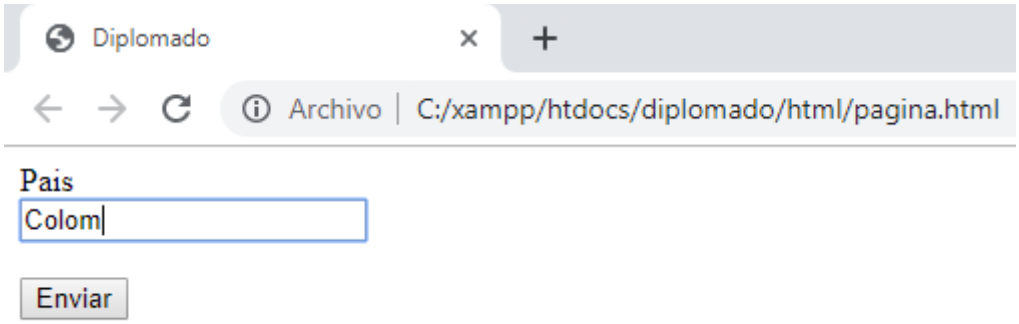




Maxlength

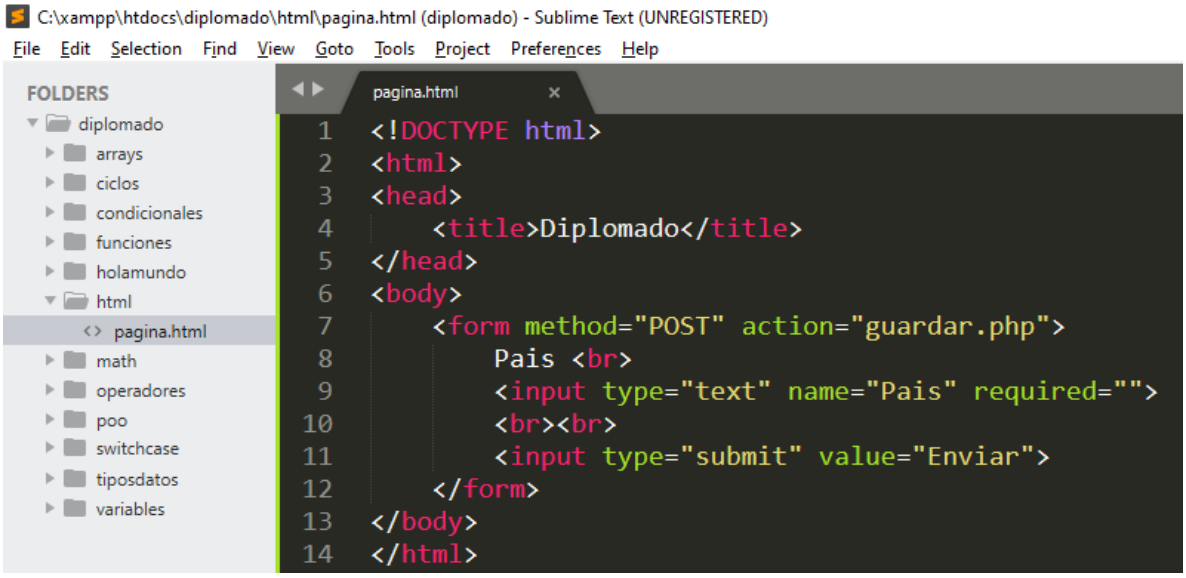
Especifica el número máximo de caracteres de un campo.

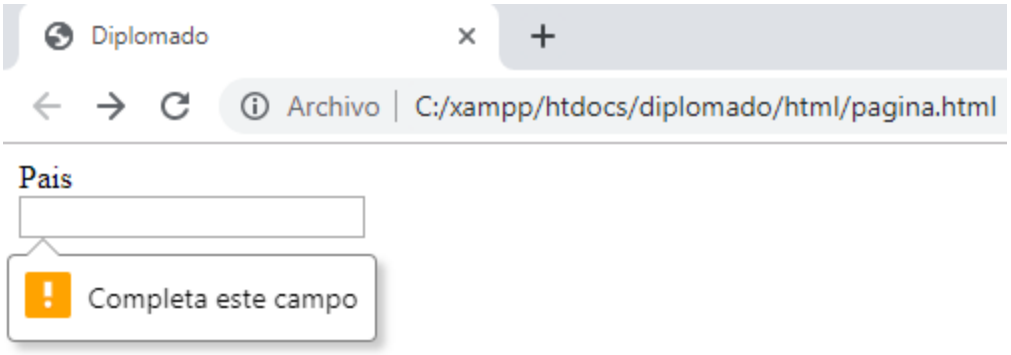




Required

Especifica que se debe completar el campo de entrada.





TEMA 8
HTML - PHP

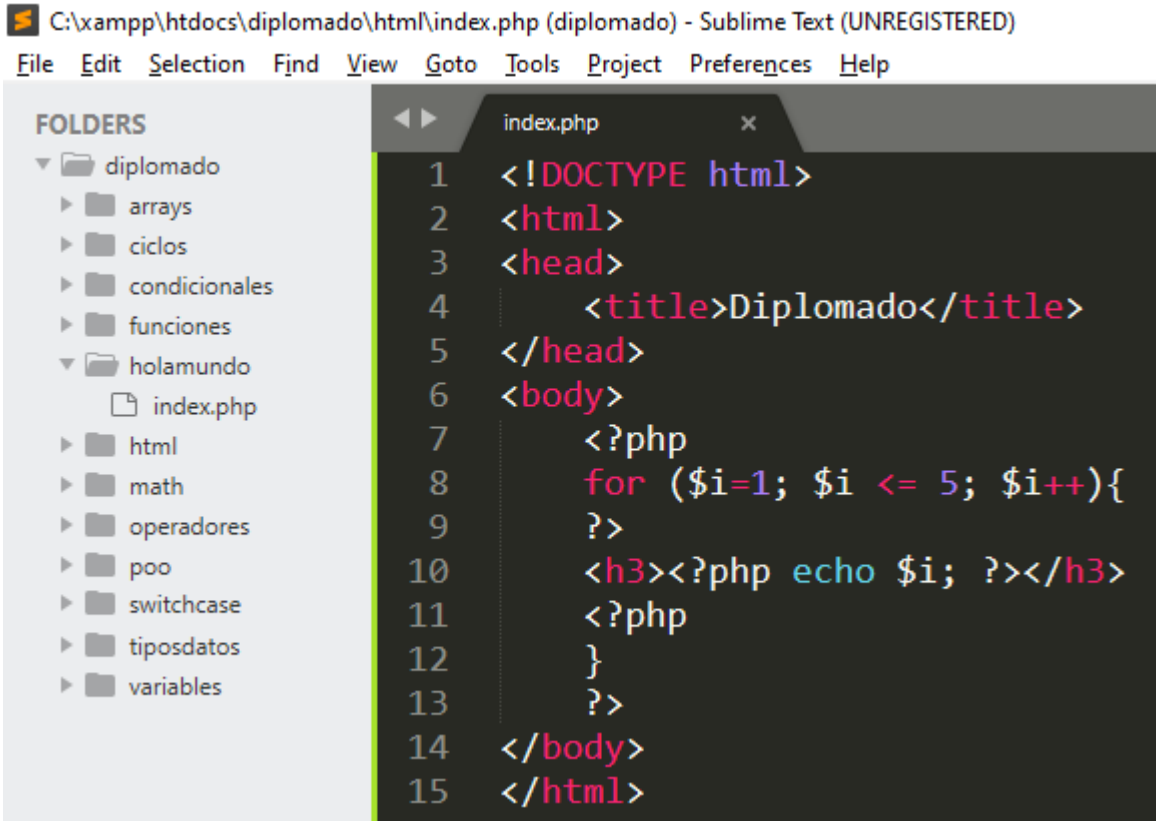
PHP es claramente un lenguaje orientado al 100% al desarrollo web, por lo cual se lleva de la mano de gran forma con el lenguaje de marcado HTML.

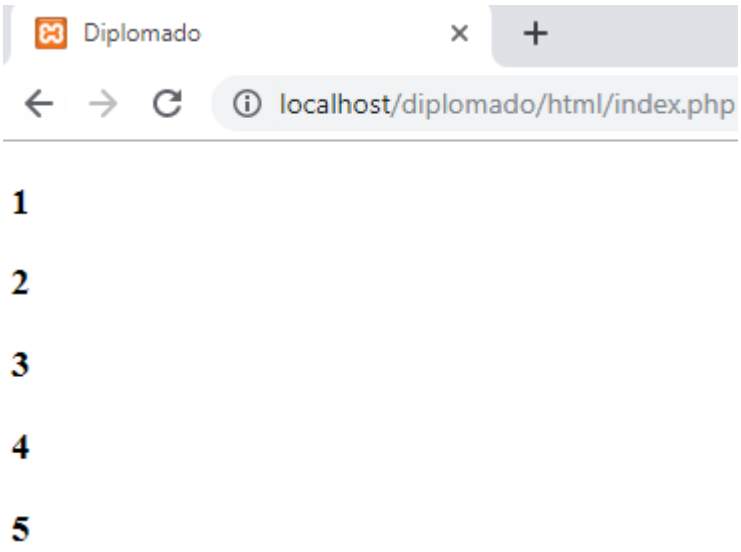
HTML y PHP se permiten integrar mutuamente en documentos donde el código de estos lenguajes es completamente legible el uno con el otro, esta es la ventaja principal de esta relación: la posibilidad de combinar código sin excepción alguna. Un ejemplo de esto podría ser el siguiente.

Generar números del 1 a 5 por medio de un ciclo y mostrar estos en encabezados <h3</h3>.

Para esto, si vienes trabajando con el archivo pagina.html, habrá que hacer un pequeño cambio, cuando estamos trabajando con PHP y HTML en un mismo documento, la extensión que debe definir el archivo es PHP, el único caso en que se puede omitir esto, son en los que en un archivo .html se tenga simplemente un formulario que envía los datos a un .php, más adelante veremos los ejemplos.

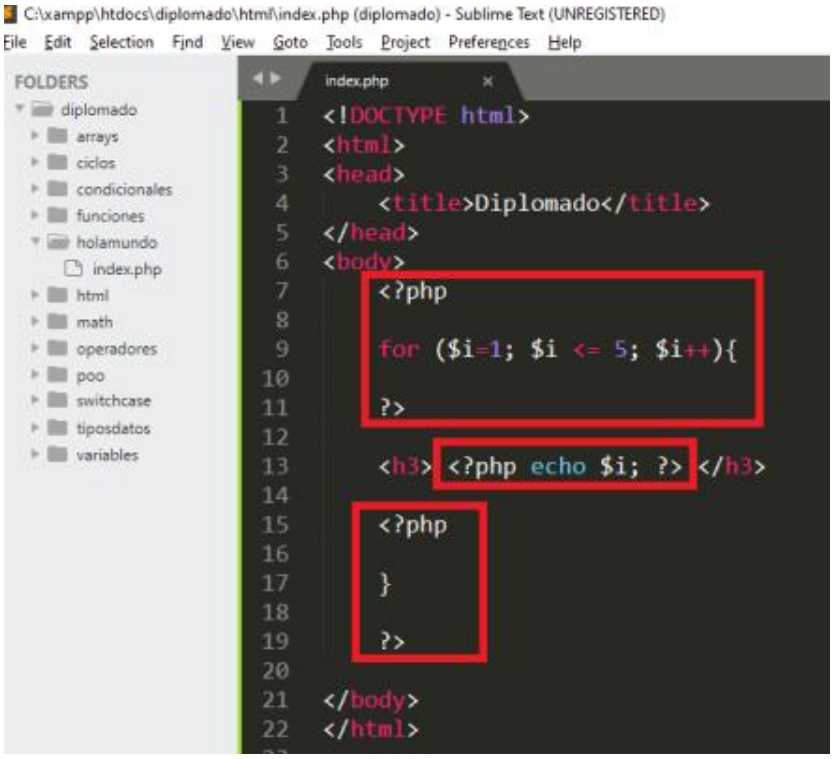
Lo ideal será continuar de ahora en adelante con todos los documentos en formato .php. Por lo que debes crear ahora un index.php.


C:\xampp\htdocs\diplomado\html\index.php (diplomado) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
diplomado
arrays
ciclos
condicionales
funciones
holamundo
index.php
html
math
operadores
poo
switchcase
tiposdatos
variables
index.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Diplomado</title>
5 </head>
6 <body>
7 <?php
8 for ($i=1; $i <= 5; $i++){
9 ?>
10 <h3><?php echo $i; ?></h3>
11 <?php
12 }
13 ?>
14 </body>
15 </html>


Diplomado
localhost/diplomado/html/index.php
1
2
3
4
5

Este es el primer acercamiento entre PHP y HTML. El contenido y etiquetas de HTML se ve perfectamente e igual en nuestro index.php. Hay que detallar es como funciona PHP y cómo se integra en el HTML.

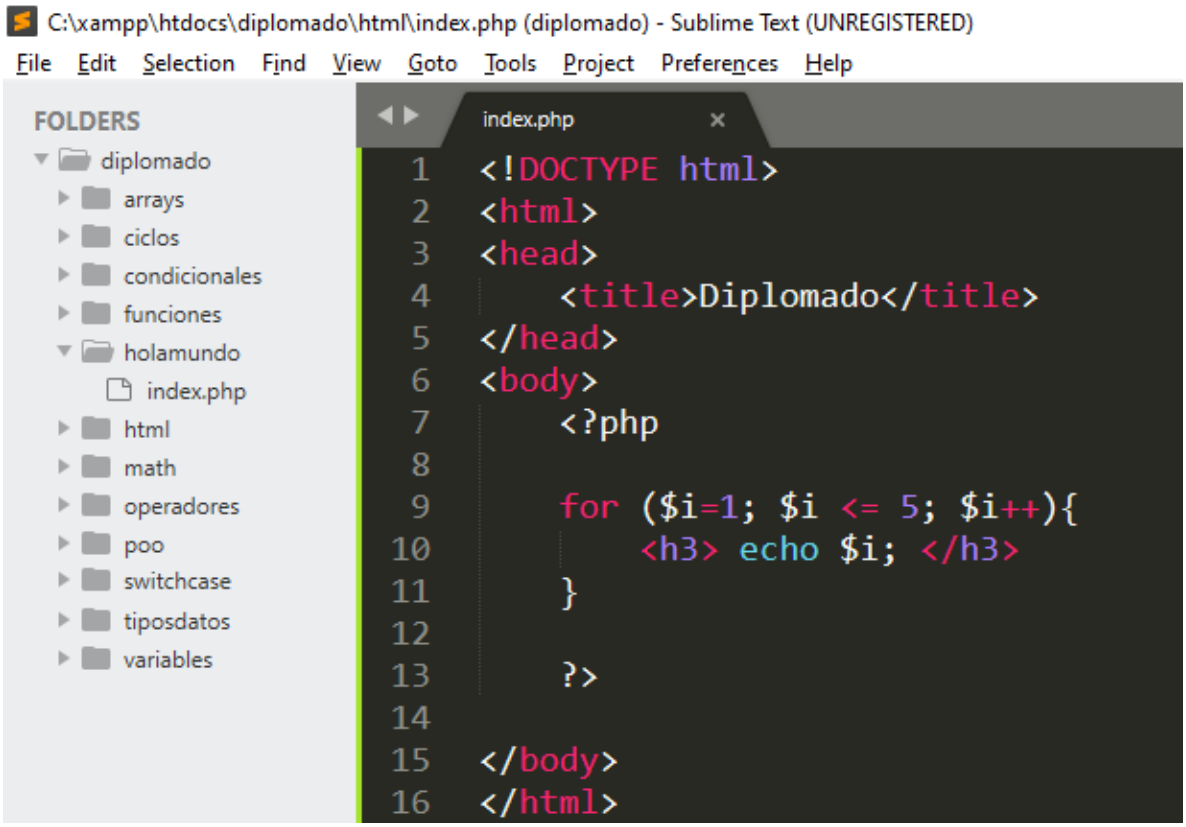
Hasta el momento, en los documentos y ejercicios realizados en PHP, solo contábamos con una etiqueta de apertura y otra de cierre (<?php ¿>). Pero tener múltiples es válido y más aún necesario cuando se trabaja con HTML, dado que se está incrustando código en varias sesiones del documento. Veamos.

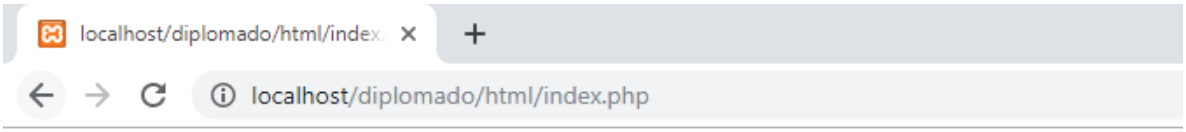


En el ejemplo anterior hay 3 casos en los que incrustamos código PHP en HTML.

- Declaración del ciclo for
- Impresión encabezado
- Cierre del ciclor for.

Esta es la forma en la que debe operar PHP frente HTML, por ejemplo, en caso de una sola declaración de etiquetas PHP y volcando todo el código dentro de estas, el resultado sería el siguiente.

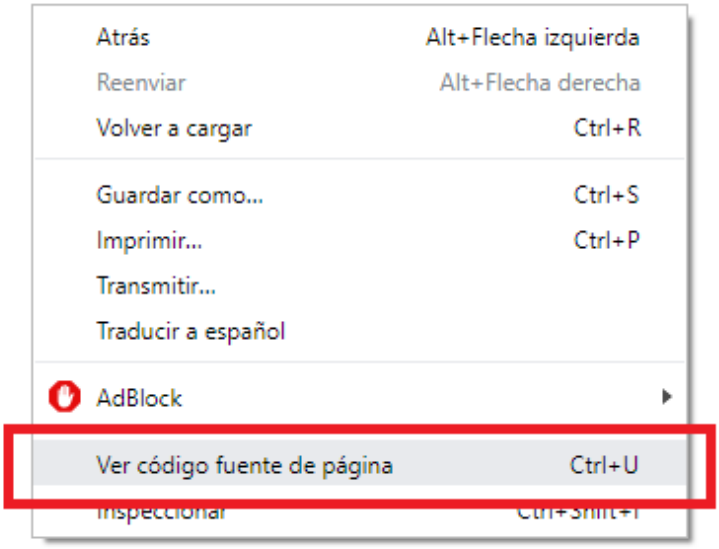


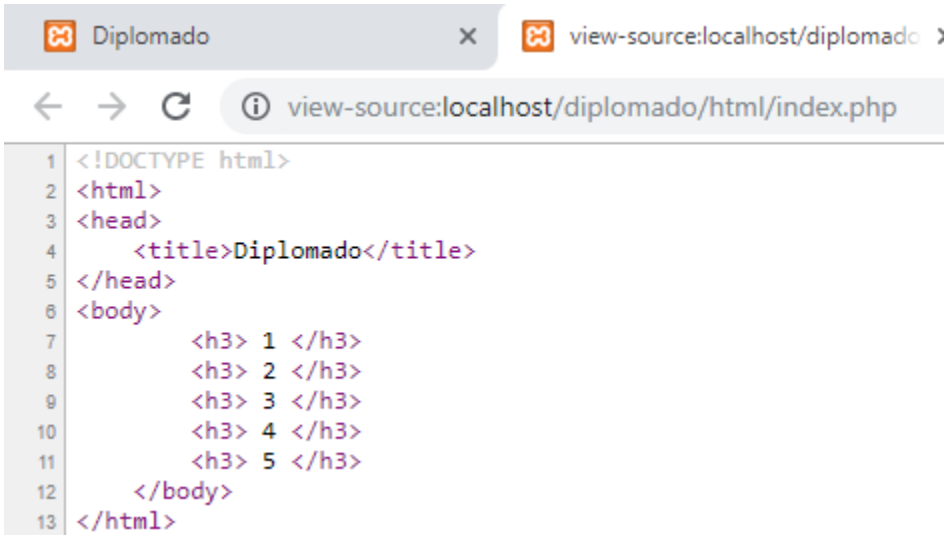


Parse error: syntax error, unexpected '<' in **C:\xampp\htdocs\diplomado\html\index.php** on line **10**

Error claramente, por eso es importante reconocer en qué lugares incrustar PHP, para eso es importante tener muy claro las etiquetas de HTML y cómo opera PHP.

Para ver en el navegador el resultado de este ejercicio (codificado de forma correcta), simplemente se debe ir al navegador y por medio de click derecho, "Ver código fuente de página"

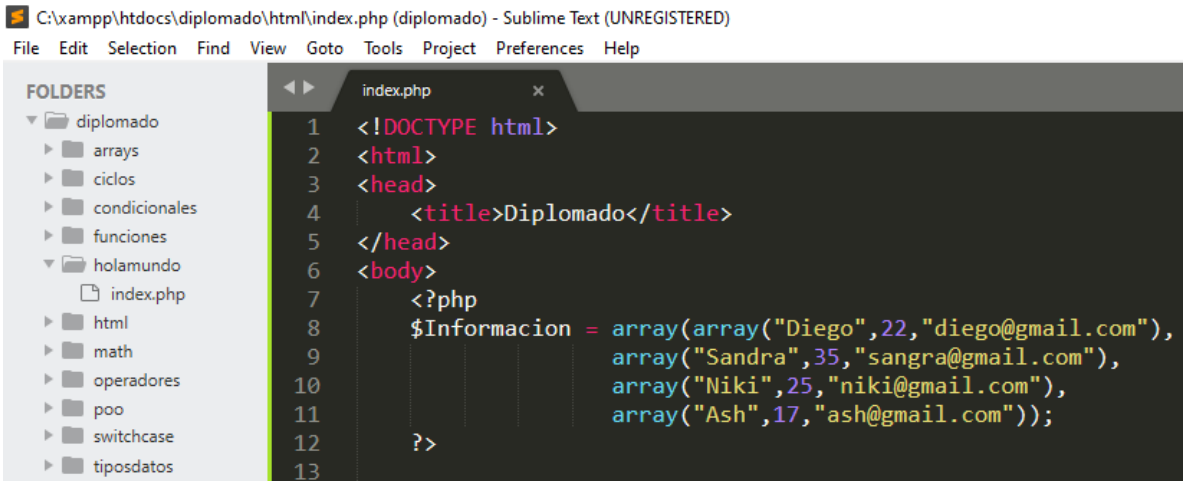




En el navegador vemos claramente el resultado en código HTML, porque esta es otra cualidad de trabajar con PHP y HTML, que, en el navegador, únicamente se visualiza el código HTML y oculta por completo cualquier pieza de código de PHP, lo cual, lo hace muy seguro al ocultar el lenguaje de programación con el que se está trabajando.

Veamos más ejemplos de código HTML en relación con PHP.

En este caso, generaremos una tabla a partir de un array con la siguiente estructura.



En este primer bloque del archivo vamos a declarar el arreglo con la siguiente información:

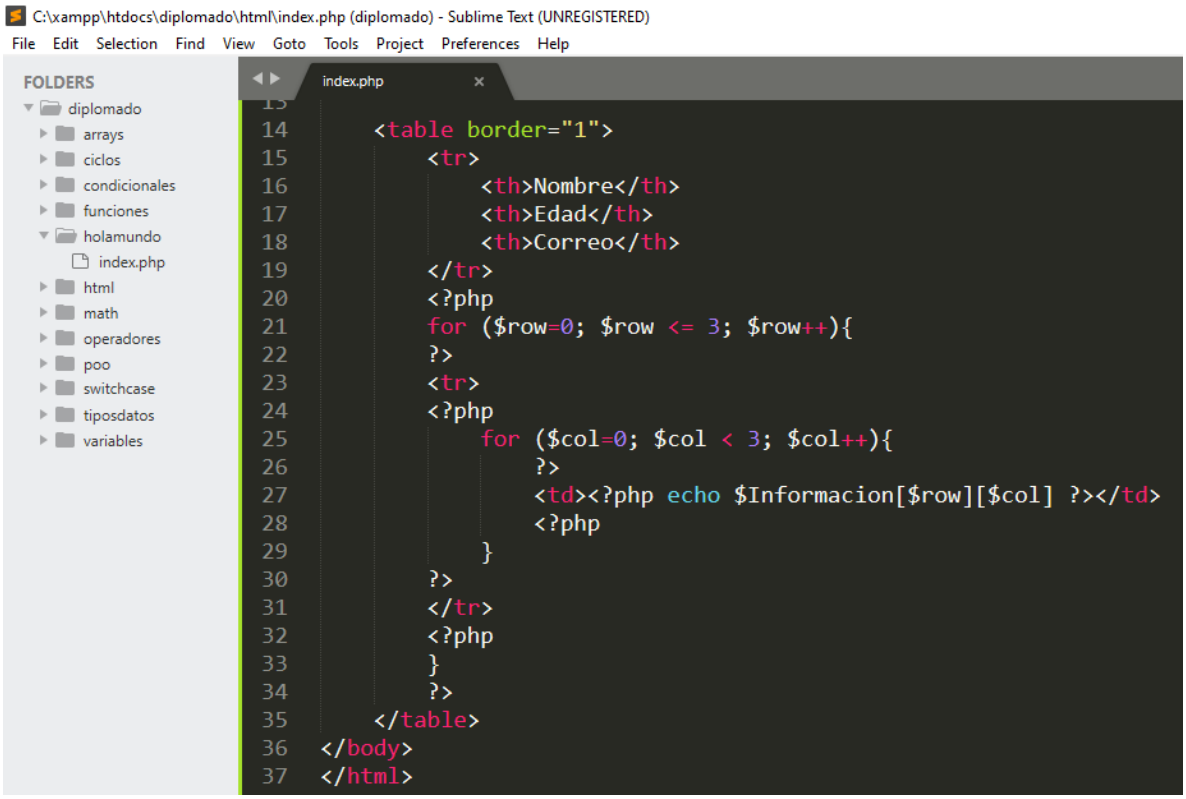
Nombre	Edad	Correo
Diego	22	diego@gmail.com
Sandra	35	sandra@gmail.com
Niki	25	niki@gmail.com
Ash	17	ash@gmail.com

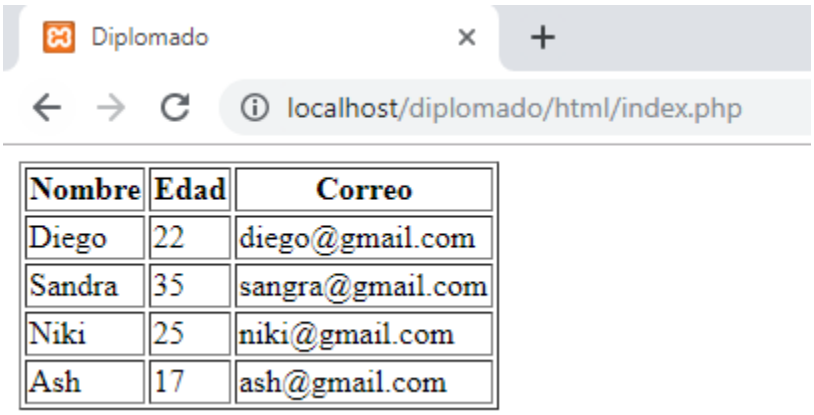
Es un arreglo multidimensional con 4 arrays internos de datos. Deseamos mostrar toda la información en una tabla, recuerda que ya aprendimos como declarar una tabla y para recorrer el array se ocupan dos ciclos for; uno para filas y otro para columnas.

Recuerda que la etiqueta <tr> se encarga de las filas y la etiqueta <td> de las columnas, es decirte, hay un ciclo para cada etiqueta.

La forma de acceder al array es por medio de dos posiciones, fila y columna, de la siguiente forma: $datos[0][2]. En este caso retornaría el tercer dato de la primera fila. (Recuerda que los índices empiezan en 0).

El resultado sería siguiente:

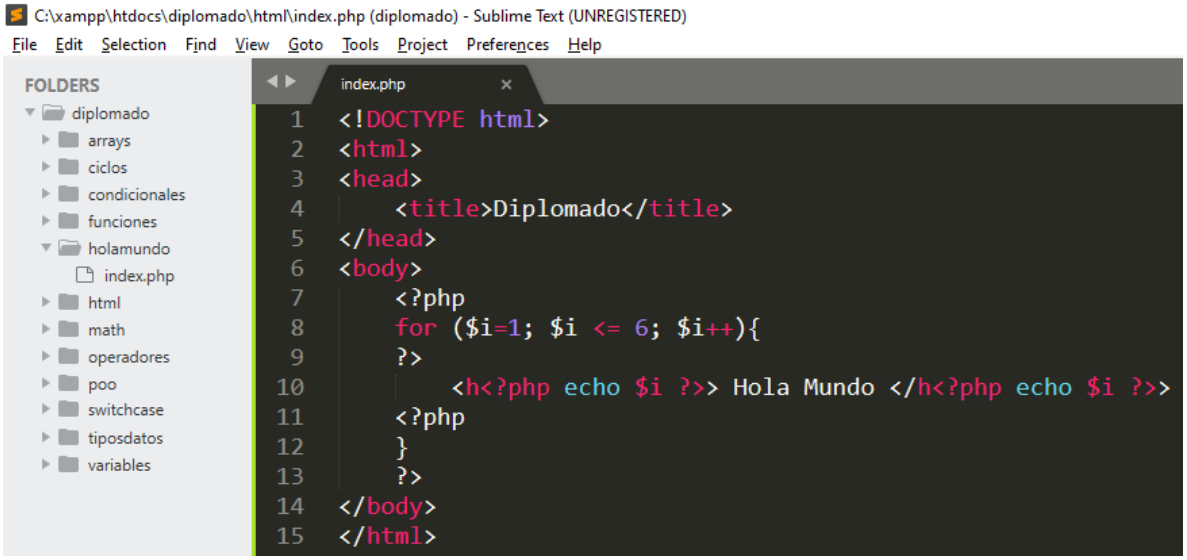




Nombre	Edad	Correo
Diego	22	diego@gmail.com
Sandra	35	sangra@gmail.com
Niki	25	niki@gmail.com
Ash	17	ash@gmail.com

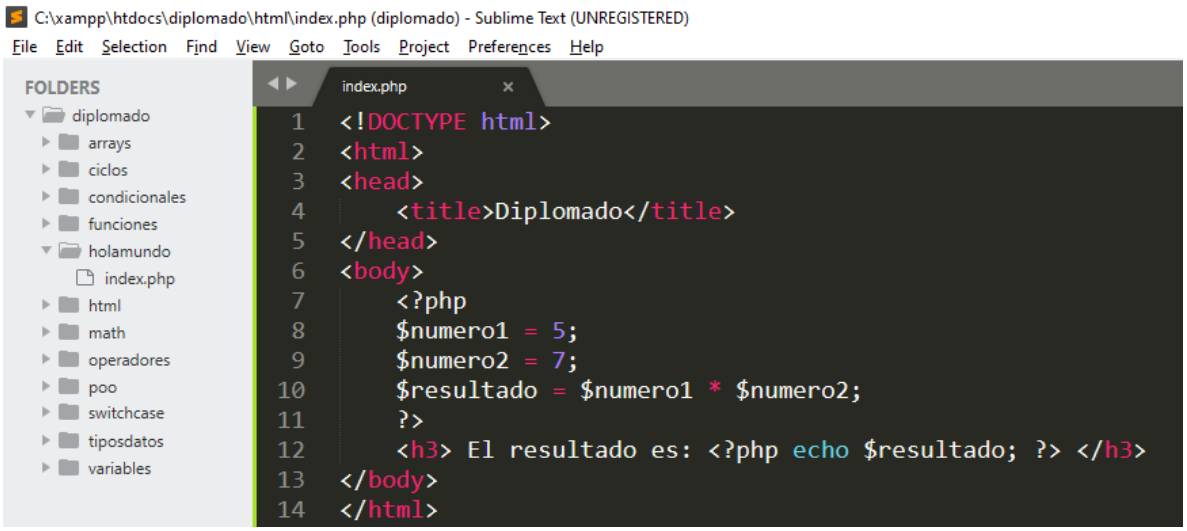
Con cada recorrido del primer ciclo de filas, creamos todos los <tr> para contener los <td> de las columnas, esto debe quedar muy claro. Lo más difícil puede ser entender cómo integrar PHP en HTML, pero con práctica será muy fácil.

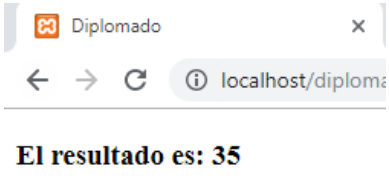
El código PHP puede ir incrustado directamente de las declaraciones de las etiquetas, por ejemplo, en los encabezados. <h1>, el uno podría remplazarse por una variable de PHP, como por ejemplo con un ciclo, e imprimir todos los encabezados, veamos.





Otro ejemplo puede ser una operación aritmética dentro de HTML e imprimiendo el resultado en un encabezado.

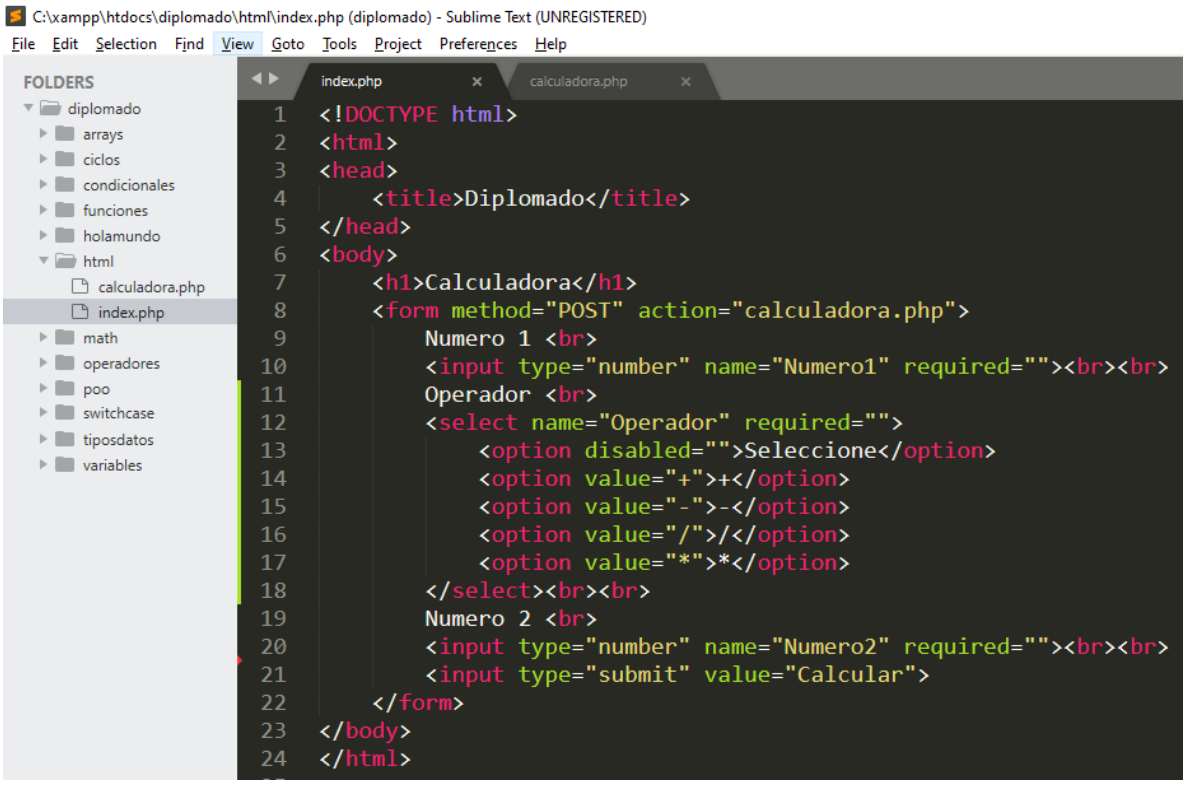


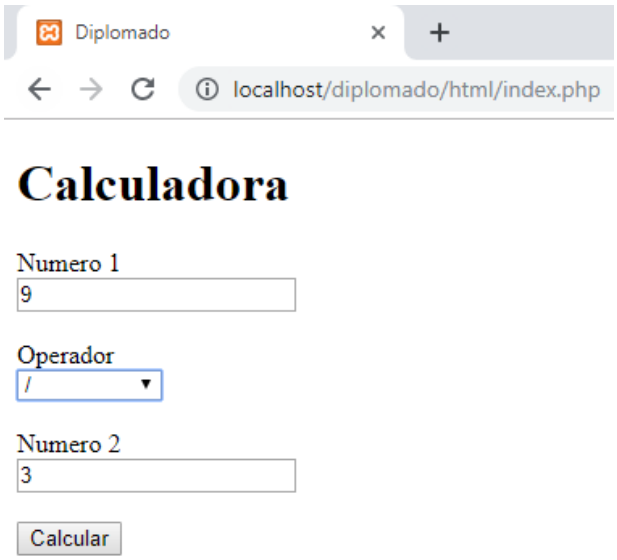


Pasemos ahora a conceptos mucho más increíbles en el desarrollo con PHP y HTML. El envío de datos por medio de formularios, vamos a usar los métodos _GET y _POST.

Para realizar el primer ejercicio, supongamos que debemos realizar una calculadora con 2 números y un operador. Los dos números deben ser campos de entrada de número, es decir, type=”number” y con la restricción de requeridos, además, el operador será una lista desplegable con los 4 (+,-,*,/). Los datos deberán ser enviados a un archivo en el folder “html” con nombre: “calculadora.php” y mostrar el resultado en este.

Primero utilizaremos el método POST.





C:\xampp\htdocs\diplomado\html\calculadora.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS
- diplomado
  - arrays
  - ciclos
  - condicionales
  - funciones
  - holamundo
  - html
    - calculadora.php
    - index.php
  - math
  - operadores
  - poo
  - switchcase
  - tiposdatos
  - variables

index.php | calculadora.php

```php
<?php

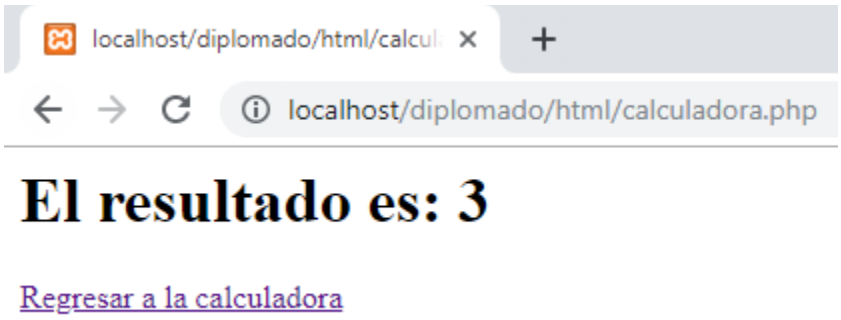
$Numero1 = $_POST['Numero1'];
$Numero2 = $_POST['Numero2'];
$Operador = $_POST['Operador'];

switch ($Operador) {
    case '+':
        $Resultado = $Numero1 + $Numero2;
        break;
    case '-':
        $Resultado = $Numero1 - $Numero2;
        break;
    case '/':
        $Resultado = $Numero1 / $Numero2;
        break;
    case '*':
        $Resultado = $Numero1 * $Numero2;
        break;
    default:
        $Resultado = "Error";
        break;
}

?>

<h1>El resultado es: <?php echo $Resultado; ?></h1>
<a href="index.php">Regresar a la calculadora</a>

```



Con pocas líneas de código construimos una pequeña calculadora que funciona de la siguiente forma.

El formulario de captura de datos cuenta con 3 campos:

- Campo número 1: Campo de type "number" para recibir únicamente números, el name es "Numero1" para recibir dos datos después, además es requerido.

```
<input type="number" name="Numero1" required=""><br><br>
```

- Campo número 2: Campo de type "number" para recibir únicamente números, el name es "Numero2" para recibir dos datos después, además es requerido.

```
<input type="number" name="Numero2" required=""><br><br>
```

- Lista desplegable: Lista desplegable con los operadores (+, -, *, /), el name es "Operador" para recibir lo datos después y además es requerido.

```
<select name="Operador" required="">
    <option disabled="">Seleccione</option>
    <option value="+">+</option>
    <option value="-">-</option>
    <option value="/">/</option>
    <option value="*">*</option>
</select><br><br>
```

- La etiqueta <form> deberá contener dos atributos muy importantes, el método y el action.

```
<form method="POST" action="calculadora.php">
```

Recordemos que el método será la forma de enviar la información, que puede ser, vía POST o GET

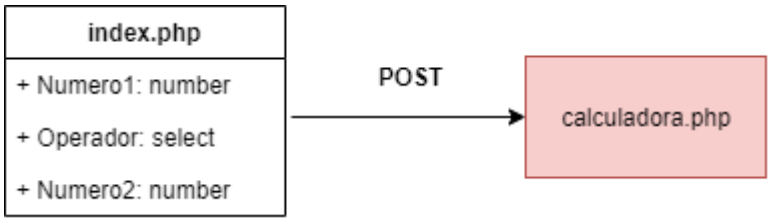
POST: envía los datos ocultos y seguros.

GET: manda los datos visibles.

POST se suele usar principalmente para los formularios, en la mayoría de los casos, es la mejor opción para los formularios por mantener la información oculta y segura, evitando la sensibilidad de la información enviada. Más adelante en el diplomado verás el uso y aplicación de cada uno de mejor forma.

GET se suele usar principalmente para enviar datos por la URL, es usado puntualmente para casos muy específicos de envío de información, por ejemplo, en buscadores, etc.

Action por otra parte, especifica el lugar donde deseamos enviar la información recolectada en los campos del formulario, el lugar en la mayoría de los casos es un archivo .php y se debe especificar muy bien la ruta del mismo exactamente.



C:\xampp\htdocs\diplomado\html\calculadora.php (diplomado) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- diplomado
  - arrays
  - ciclos
  - condicionales
  - funciones
  - holamundo
  - html
    - calculadora.php
    - index.php
  - math
  - operadores
  - poo
  - switchcase
  - tiposdatos
  - variables

index.php | calculadora.php

```php
<?php

$Numero1 = $_POST['Numero1'];
$Numero2 = $_POST['Numero2'];
$Operador = $_POST['Operador'];

switch ($Operador) {
    case '+':
        $Resultado = $Numero1 + $Numero2;
        break;
    case '-':
        $Resultado = $Numero1 - $Numero2;
        break;
    case '/':
        $Resultado = $Numero1 / $Numero2;
        break;
    case '*':
        $Resultado = $Numero1 * $Numero2;
        break;
    default:
        $Resultado = "Error";
        break;
}

?>

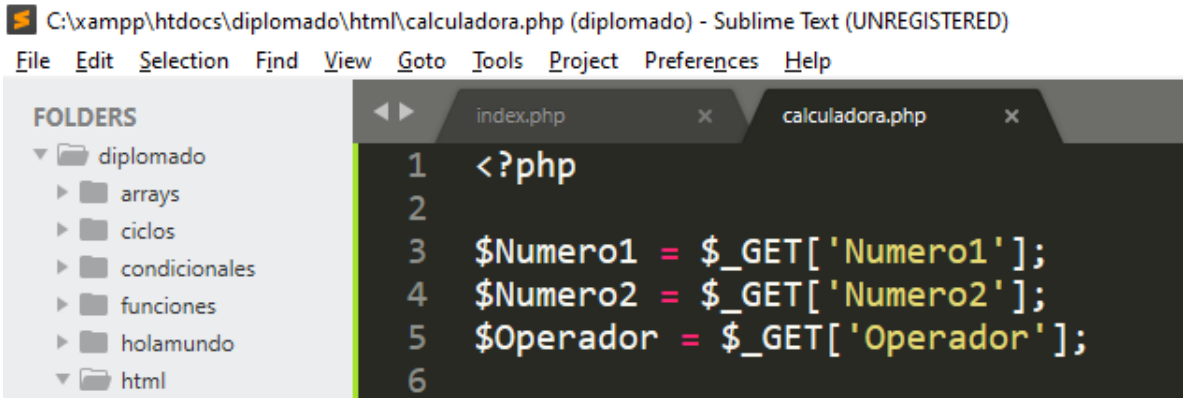
<h1>El resultado es: <?php echo $Resultado; ?></h1>
<a href="index.php">Regresar a la calculadora</a>
```

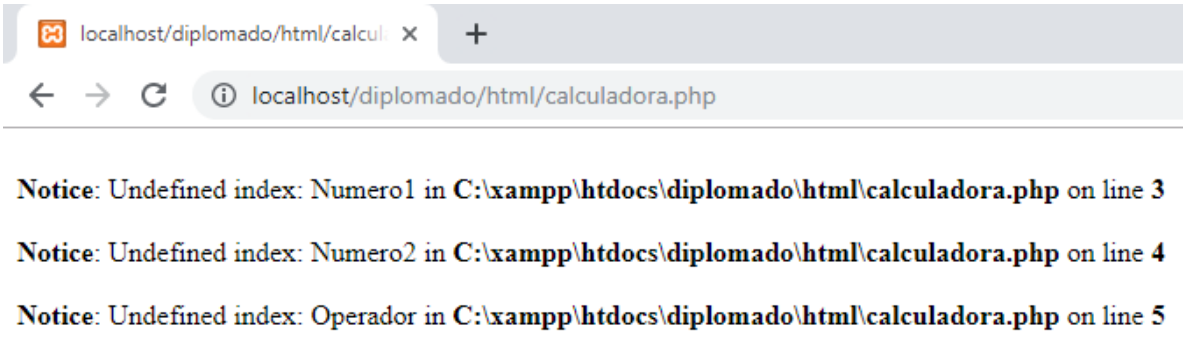
El archivo calculador .php se encargará en primera instancia en recibir los datos enviados por desde el formulario, de la siguiente forma.

Cuando se define el método POST en el formulario en sus atributos, ese será el método sobre el que seba a enviar la información:

- Si se envía por POST, se recibe por POST.
- Si se envía por GET, se recibe por GET.

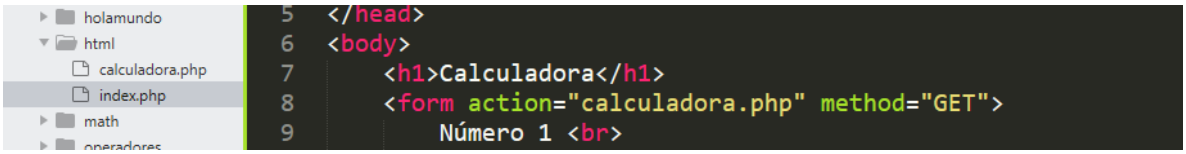
Una prueba sencilla de esto es, cambiar en calculadora.php la forma de recibir la información, de POST a GET y veremos el error al enviar una operación.

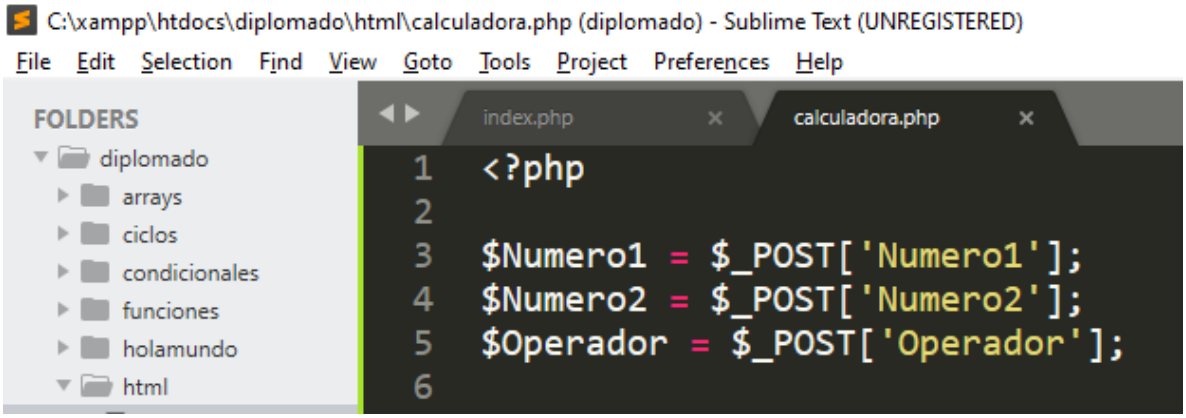


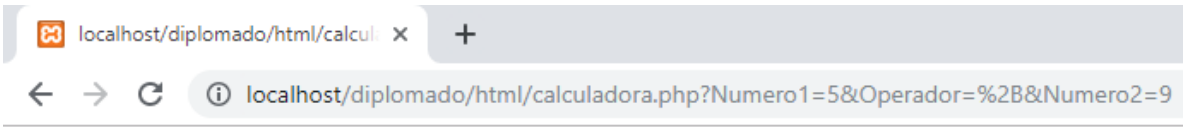


El error radica en que PHP no encuentra las 3 variables ($Numero1, $Numero2, $Operador) en enviadas con un método $_GET.

Lo mismo ocurre si envío por GET y recibo por POST. Veamos.







Notice: Undefined index: Numero1 in **C:\xampp\htdocs\diplomado\html\calculadora.php** on line **3**

Notice: Undefined index: Numero2 in **C:\xampp\htdocs\diplomado\html\calculadora.php** on line **4**

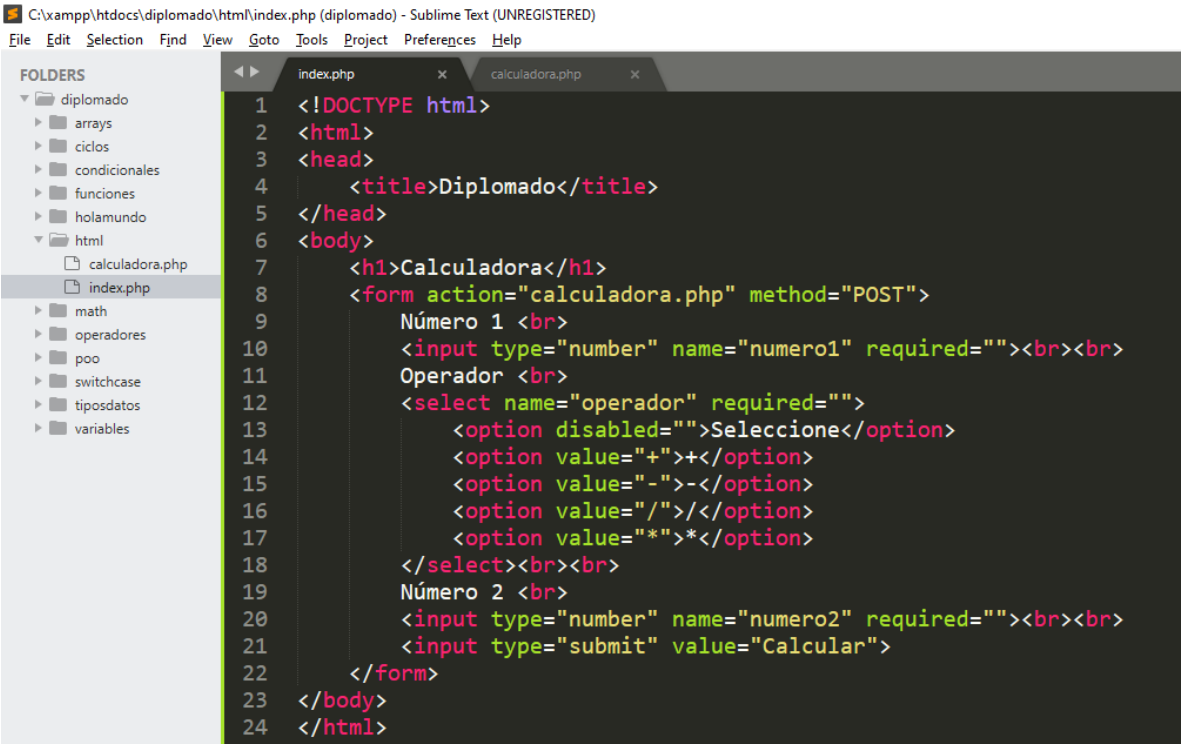
Notice: Undefined index: Operador in **C:\xampp\htdocs\diplomado\html\calculadora.php** on line **5**

El resultado es: Error

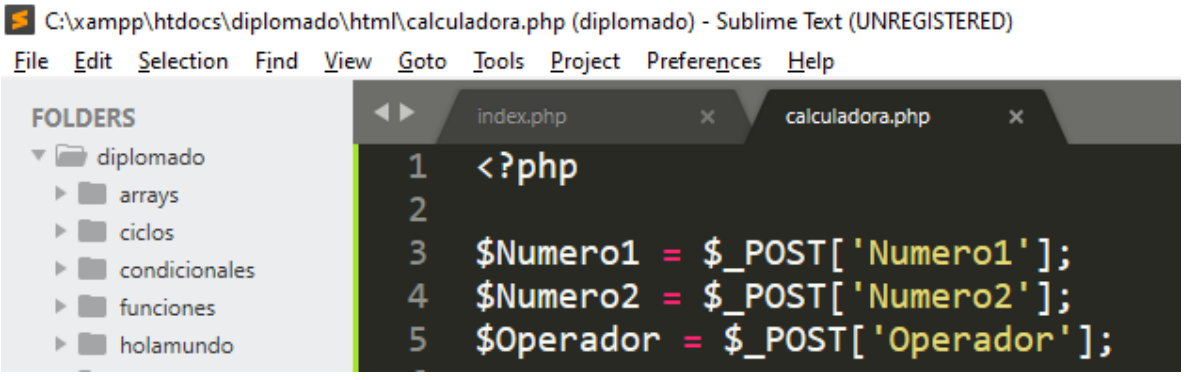
Regresar a la calculadora

El mismo error. Por eso es muy importante la consistencia de la información en relación con los métodos de envío y recepción de los datos. Continuemos con la explicación de la calculadora.

Otro error muy común es la incosistencia entre los name="" y los métodos al recibir la información, por ejemplo:



Todos los atributos name de los campos de entrada han sido cambiados, ahora todos tienen el mismo estilo, todo en minúscula. Veamos ahora el archivo calculadora.php.



Los métodos de envío coinciden en ambos casos, envío por POST, recibo por POST. Pero no hay consistencia en el nombre de los atributos, veamos que ocurre cuando enviamos información.



Exactamente el mismo error, en conclusión, los dos principales aspectos en uso de los formularios radican en:

- Métodos de envío y recepción de información.
- Nombre de los campos de entradas.

Teniendo la información encapsulada en nuestras variables después de cada envío del formulario, se puede operar y determinar el resultado, una de las formas más fácil de hacer es por medio de la estructura Switch. Evaluando cada caso a raíz de la operación seleccionada.

```php
switch ($Operador) {
    case '+':
        $Resultado = $Numero1 + $Numero2;
        break;
    case '-':
        $Resultado = $Numero1 - $Numero2;
        break;
    case '/':
        $Resultado = $Numero1 / $Numero2;
        break;
    case '*':
        $Resultado = $Numero1 * $Numero2;
        break;
    default:
        $Resultado = "Error";
        break;
}
```

Recuerda el módulo número dos y la explicación del mismo concepto.

En la variable resultado vamos a contener el resultado de la operación según la seleccionada por el usuario y los números ingresados.

Finalmente haremos uso de dos conceptos directamente de HTML: Encabezados e Hipervínculos.

```
26
27  <h1>El resultado es: <?php echo $Resultado; ?> </h1>
28  <a href="index.php">Regresar a la calculadora</a>
29
```

El resultado estará en un <h1> y tendremos un hipervínculo para regresar a la calculadora. Ejercicio finalizado.

Otro ejercicio será simular un formulario de registro como Facebook. Veamos el formulario.

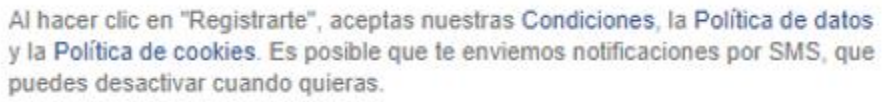
Formulario de Registro:



En este formulario existen 11 campos de entrada de información (Contando el botón), divididos de la siguiente forma.

- Campo de entrada de texto para el nombre.
  <input type="text">
- Campo de entrada de texto para el apellido.
  <input type="text">
- Campo de entrada de email para el correo electrónico.
  <input type="email">
- Campo de entrada de password para la contraseña.
  <input type="password">
- Lista desplegable para el día del nacimiento.
  <select>
- Lista desplegable para el mes de nacimiento.
  <select>
- Lista desplegable con el año de nacimiento.
  <select>
- Campo de entrada de selección para el sexo: "Mujer".
  <input type="radio">
- Campo de entrada de selección para el sexo: "Hombre".
  <input type="radio">
- Campo de entrada de selección para el sexo: "Personalizado".
  <input type="radio">
- Botón de envío de información.
  <input type="submit">

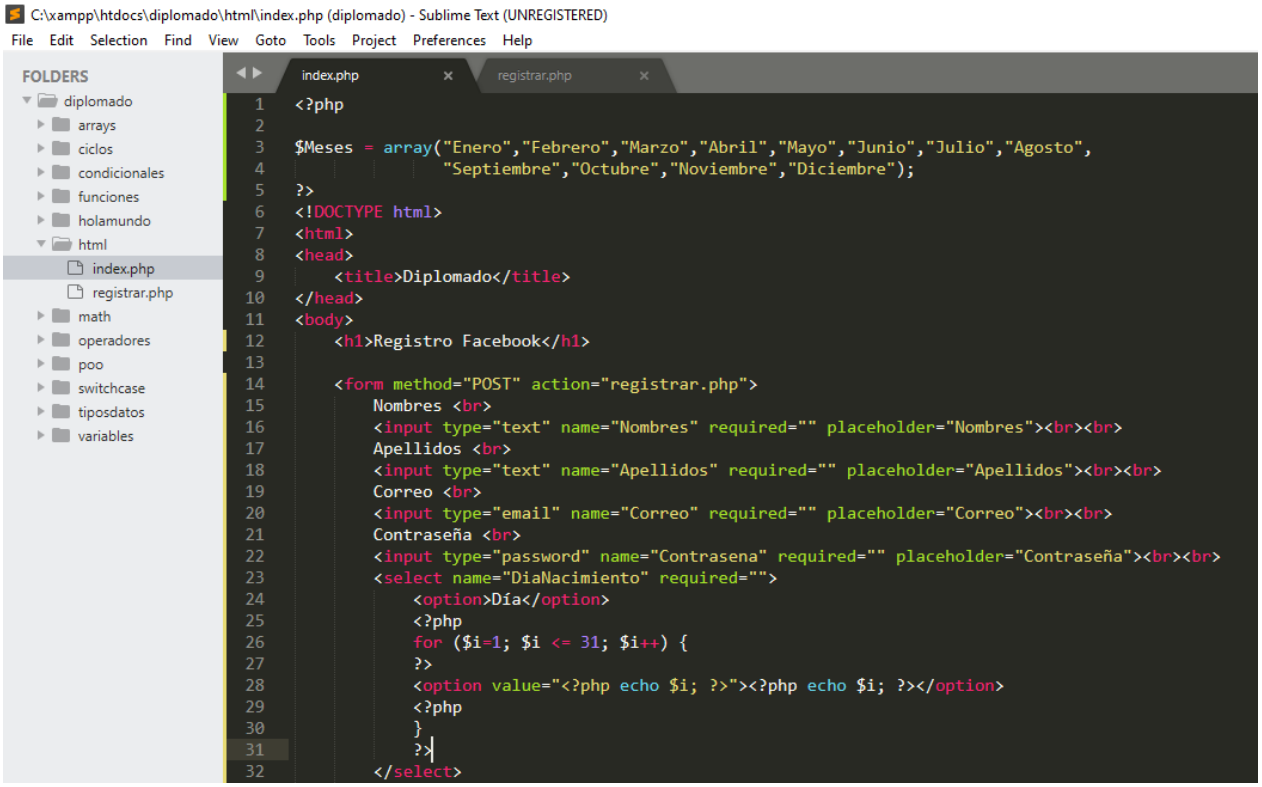
Adicionalmente replicaremos el párrafo de información adicional del formulario.



Por otra parte, todos los atributos name tiene que ser claros:

- Nombres.
- Apellidos.
- Correo.
- Contrasena. (Recuerda que la ñ no es válida).
- DiaNacimiento.
- MesNacimiento.
- AnoNacimiento (Recuerda que la ñ no es válida).
- Sexo. (Recuerda que los tres type=”radio”, deben contar con el mismo nombre dado que la selección de sexo se da gracias a estas tres opciones.

Veamos ahora el formulario codificado y con sus respectivos campos de entrada declarados, nombrados y listos para enviar información.





- Para ver el resultado da click aquí**:** **Formulario de Registro**
- Para descargar el archivo da click aquí: **Descargar**

Este formulario cuenta con un gran volumen de datos por lo que hay que ser cuidados en manejar el diseño y funcionalidad del mismo.

La forma en la que opera es la siguiente:

```
<form method="POST" action="registrar.php">
```

- El método de envío es POST.
- El archivo al enviar la información será **"**registrar.php**".**

```
Nombres <br>
<input type="text" name="Nombres" required="" placeholder="Nombres"><br><br>
Apellidos <br>
<input type="text" name="Apellidos" required="" placeholder="Apellidos"><br><br>
Correo <br>
<input type="email" name="Correo" required="" placeholder="Correo"><br><br>
Contraseña <br>
<input type="password" name="Contrasena" required="" placeholder="Contraseña"><br><br>
```

- Los campos de entrada, cada uno cuenta con las siguientes características:
  1. Atributo type especificado.
  2. Atributo name claro.
  3. Atributo required.
  4. Atributo placeholder con la descripción del campo.

El atributo placeholder es nuevo, este permite definir un texto, predeterminado en los campos de entrada. El texto es simplemente visual, no se encuentra habilitado como un valor dentro del campo.

La forma más óptima de mostrar los días del mes en la lista desplegable es por medio de un ciclo for para recorrer todos los días de un mes. Comprendiendo los días del mes desde 1 hasta 31, de la siguiente forma.

```
<select name="DiaNacimiento" required="">
    <option>Día</option>
    <?php
    for ($i=1; $i <= 31; $i++) {
    ?>
    <option value="<?php echo $i; ?>"><?php echo $i; ?></option>
    <?php
    }
    ?>
</select>
```

Para escribir los doce meses del año, para evitar escribir uno por uno en la lista desplegable (también es válido hacerlo), cree un array con los doce meses y en la lista desplegable se ejecuta un foreach para mostrar los mismo.

```
<?php

$Meses = array("Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto",
               "Septiembre","Octubre","Noviembre","Diciembre");
?>
```

```
<select name="MesNacimiento" required="">
    <option>Mes</option>
    <?php
    foreach ($Meses as $Mes) {
    ?>
    <option value="<?php echo $Mes; ?>"><?php echo $Mes; ?></option>
    <?php
    }
    ?>
</select>
```

La forma que se implementó para obtener los años en la lista desplegable es una muy recursiva, un ciclo for que se decrementa a razón de uno (un año a la vez) hasta llegar a un límite, en este caso hasta 1950 (2019-1950).

```
<select name="AnoNacimiento" required="">
    <option>Año</option>
    <?php
    for ($i=2019; $i >= 1950; $i--) {
    ?>
    <option value="<?php echo $i; ?>"><?php echo $i; ?></option>
    <?php
    }
    ?>
</select><br><br>
```

Los tipos radio son muy particulares, dado que se crean X a partir de un solo name, es decir, a partir del atributo name pueden crear los siguientes tipos, por ejemplo, para un examen, las posibles respuestas podría ser:

- **Si**
- **No**
- **Casi nunca**
- **Nunca**

Y todos contener el mismo nombre, por decir uno, name=”respuesta”, lo que permite esto es la selección unitaria de uno de esto campos, así como ocurre en el caso del formulario de ejemplo con el sexo.

```
Sexo <br>
<input type="radio" name="Sexo" value="Mujer"> Mujer
<input type="radio" name="Sexo" value="Hombre"> Hombre
<input type="radio" name="Sexo" value="Personalizado"> Personalizado
```

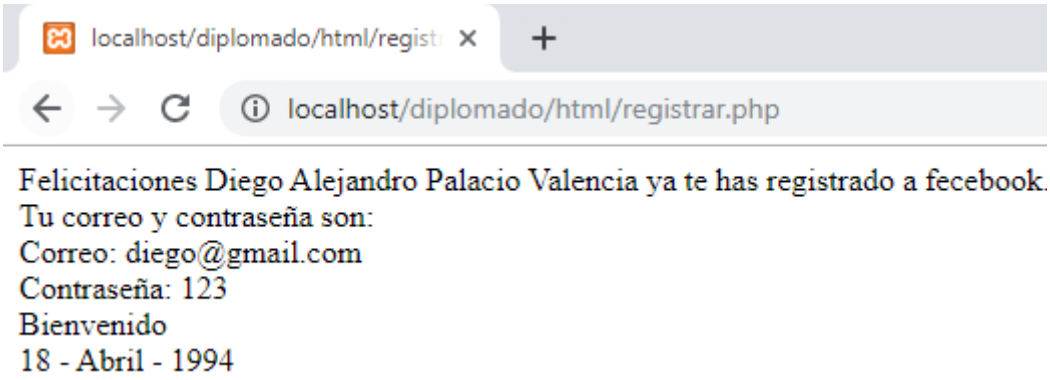
Tres diferentes campos, diferentes opciones, pero mismo atributo name.

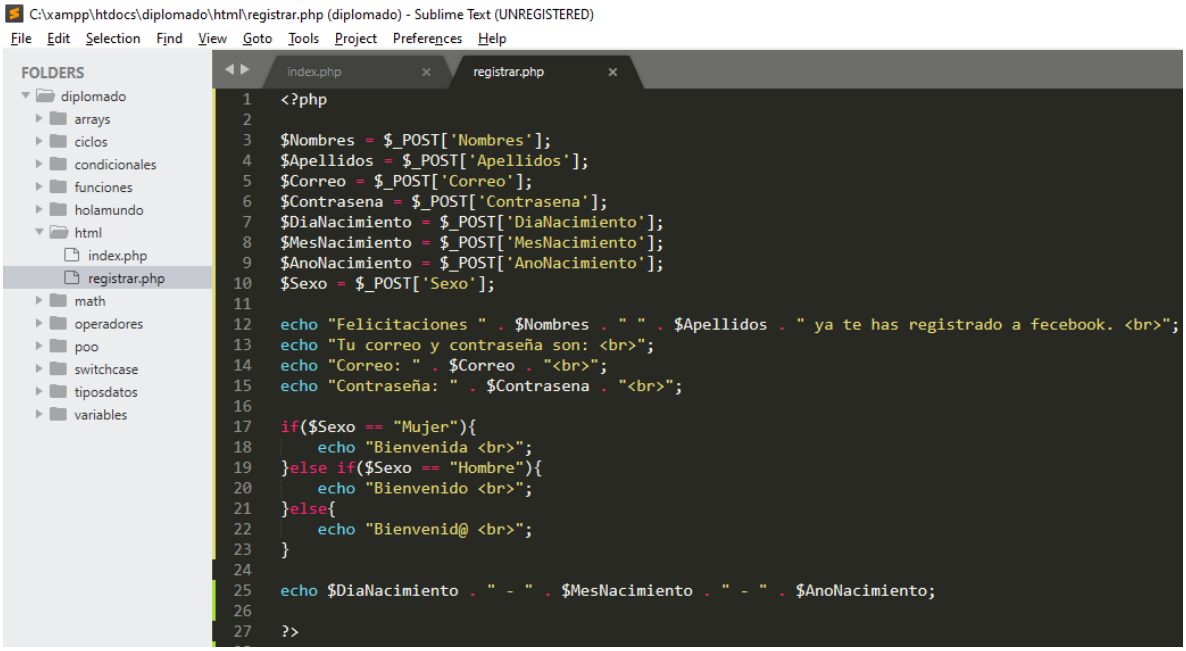
En este ejercicio conocimos una nueva etiqueta <p>, que es muy simple y funcional, permite definir un párrafo:

```
<p>Al hacer clic en "Registrarte", aceptas nuestras Condiciones, la Política de datos y la Política
de cookies.<br>Es posible que te enviemos notificaciones por SMS, que puedes desactivar cuando quieras.</p>
```

Con toda la información de los campos completada, simplemente resta realizar el envío al action descrito, en este caso, registrar.php.

Recibiremos todos los campos enviados por POST para procesarlos debidamente y mostrar la siguiente estructura:





Lo primero que se debe hace en este caso es siempre recibir la información enviada por el método definido, sea GET o POST:

```php
$Nombres = $_POST['Nombres'];
$Apellidos = $_POST['Apellidos'];
$Correo = $_POST['Correo'];
$Contrasena = $_POST['Contrasena'];
$DiaNacimiento = $_POST['DiaNacimiento'];
$MesNacimiento = $_POST['MesNacimiento'];
$AnoNacimiento = $_POST['AnoNacimiento'];
$Sexo = $_POST['Sexo'];
```

Recuerda siempre que los atributos name deben coincidir con las variables recibidas en $_POST[], deben ser claros y dicientes.

Los primeros echos se encargan de hacer uso de las variables recibidas, este proceso es completamente similar al visto en los módulos anteriores cuando realizábamos impresión de variables con echo y texto plano. (No olvides del etiqueta de salto de línea <br>).

```
echo "Felicitaciones " . $Nombres . " " . $Apellidos . " ya te has registrado a fecebook. <br>";
echo "Tu correo y contraseña son: <br>";
echo "Correo: " . $Correo . "<br>";
echo "Contraseña: " . $Contrasena . "<br>";
```

De nuevo haremos uso de una estructura condicional, if – else. Queremos mostrar un mensaje a partir del sexo seleccionado en el formulario.

- Bienvenida – Para las mujeres.
- Bienvenido – Para los hombres.
- Bienvenid@ - Para los personalizado.

```
if($Sexo == "Mujer"){
    echo "Bienvenida <br>";
}else if($Sexo == "Hombre"){
    echo "Bienvenido <br>";
}else{
    echo "Bienvenid@ <br>";
}
```

Y para finalizar, debemos imprimir la fecha de nacimiento concatenando dia, mes y año, de la siguiente forma:

```
echo $DiaNacimiento . " - " . $MesNacimiento . " - " . $AnoNacimiento;
```

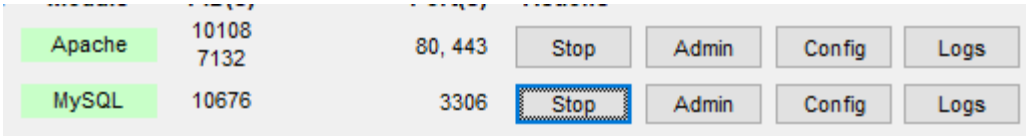
Ya hemos realizado un formulario con buena cantidad de campos y está completamente funciona. Recuerda muy bien cada uno de estos conceptos dado que son los que siempre estarán presente en el desarrollo de HTML y PHP. Puedes prácticar creando un formulario de inicio de sesión, un formulario de contacto, etc. Las posibilidades son infinitas.

Con el proyecto que desarrollaremos, realizaremos varios formularios con funcionalidades diferentes, utilizaremos tablas y bases

de datos, será básico, pero con muy contenido práctico para entender cómo funciona el desarrollo web con PHP. Más adelante en el módulo tendrás todos los requisitos y características del proyecto.

TEMA 9
Bases de Datos - MySQL

Al momento de descargar XAMPP, este ofrecía dos características fundamentales en el desarrollo bajo PHP. El primero fue Apache, el servidor que se ha utilizado a lo largo del diplomado para ejecutar nuestro código PHP en el navegador, y ahora el otro, MySQL o en otras palabras, phpMyAdmin:



phpMyAdmin es una herramienta gratuita, que permite de una manera muy completa acceder a todas las funciones de la base de datos MySQL a través de PHP, mediante una interfaz web muy intuitiva.

¿Qué ofrece phpMyAdmin?

Esta herramienta es muy completa y nos ofrece una gran cantidad de usos y características, algunas de ellas son:

- Esta aplicación nos permitirá realizar las operaciones básicas en base de datos MySQL, como son: crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar sentencias SQL, administrar claves de campos, administrar privilegios y exportar datos en varios formatos. La función de exportar datos se emplea muchas veces para realizar backups de la base de datos y poder restaurar esta copia de seguridad en el futuro a través de phpMyAdmin mediante la opción "importar".

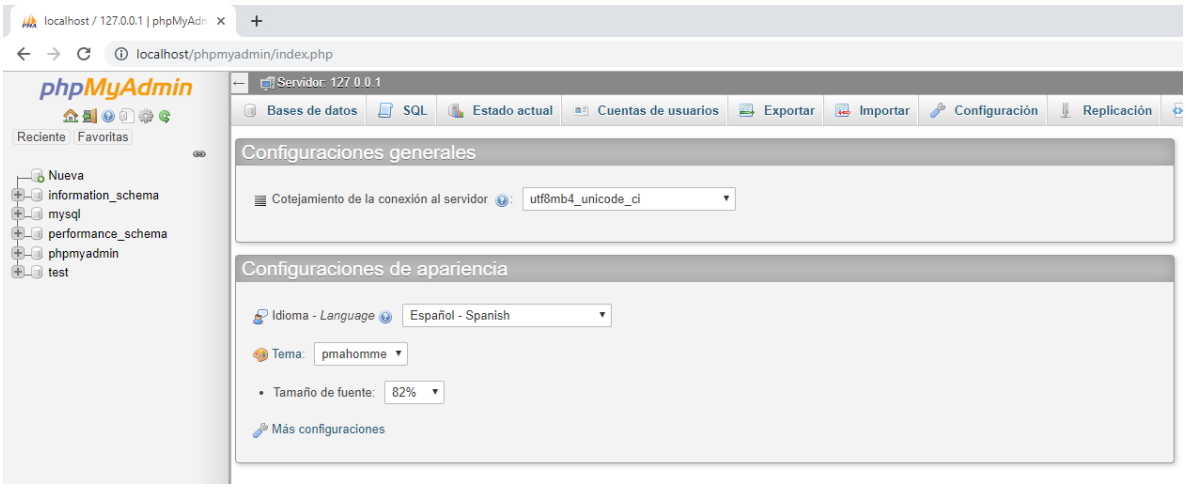
- phpMyAdmin es el administrador de bases de datos por defecto en muchos paneles de control web comerciales como son cPanel, Plesk o DirectAdmin.

- Los usuarios no deberían tener problemas a la hora de manejar esta herramienta, ya que es fácil de usar.

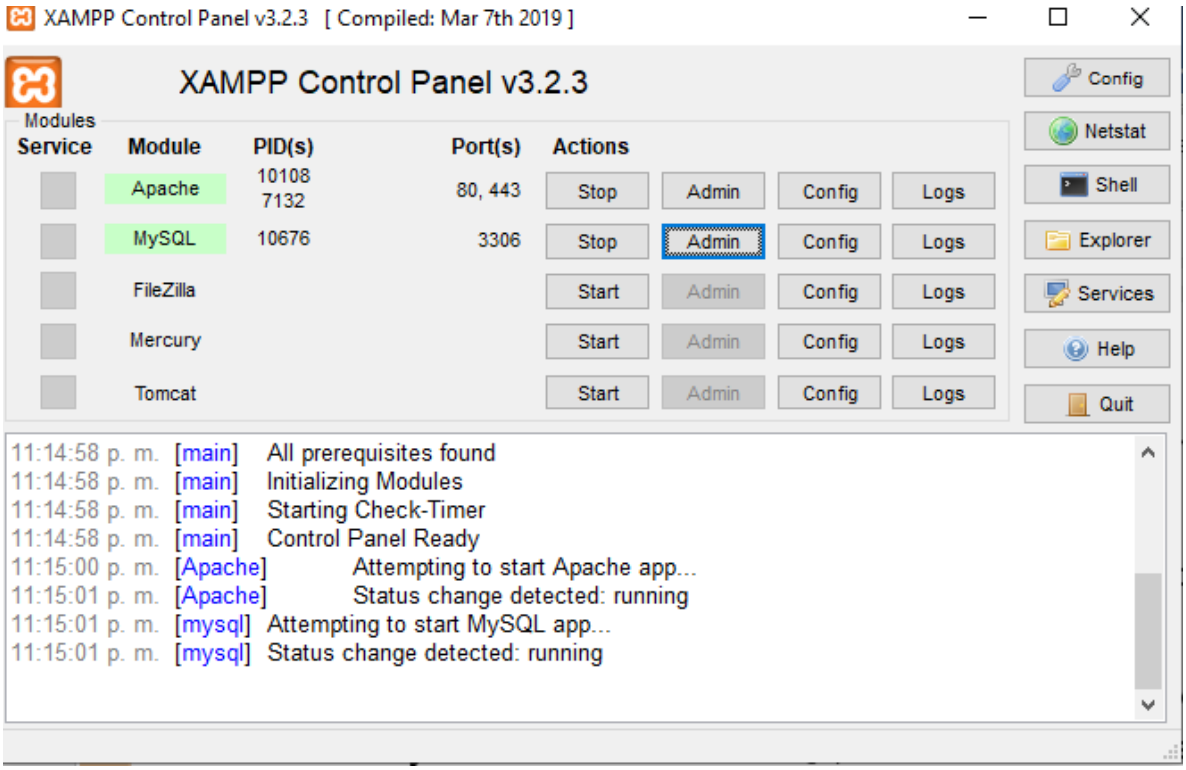
- Otra de las funciones más importantes que nos ofrece es que permite optimizar y reparar tablas, las cuales son dos tareas de mantenimiento fundamentales.

- Nos da la posibilidad de realizar búsquedas en las bases de datos, además de poder escribir nuestras propias consultas SQL de manera directa y ejecutarlas.

- Esta herramienta también es de gran ayuda para desarrolladores de aplicaciones que empleen MySQL, ya que permite depurar consultas y hacer tests de forma rápida y sencilla.



Las formas de acceder a phpMyAdmin son muy sencillas, la primera de estas es directamente por medio de XAMPP, dando click en la opción admin:



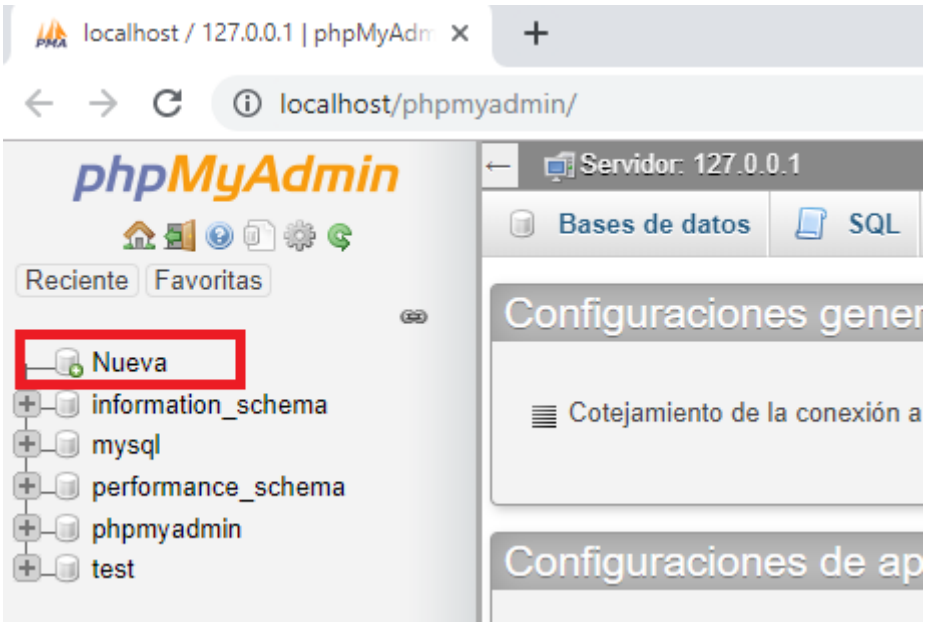
O yendo directamente a la ruta donde se ejecuta phpMyAdmin en: http://localhost/phpmyadmin/index.php.

Esta herramienta provista por XAMPP tiene una gran cantidad de características y funciones esenciales descritas anteriormente, pero el alcance del diplomado estará relacionado con los siguientes temas:

- Creación de bases de datos.
- Creación de tablas.
- Especificación de campos de tabla.
- Tipos de datos.
- Claves primarias.
- Inserción de datos.
- Consulta de datos.
- Actualización de datos.
- Eliminación de datos.
- Exportación de datos y tablas.
- Copias de seguridad.
- Modelo Entidad Relación.

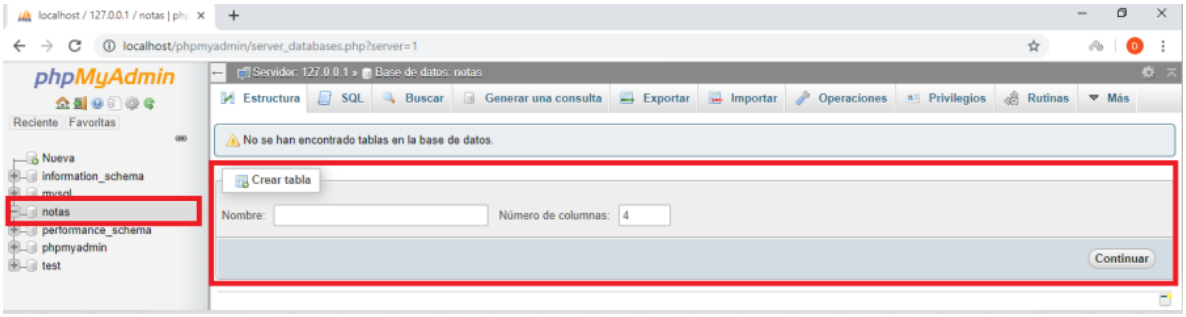
Para crear nuestra primera base de datos, que será la utilizada en el proyecto final, el proceso que debemos realizar es el siguiente.

- Abrir phpMyAdmin.
- En el apartado superior izquierdo, se encuentran las bases de datos establecidas por defecto de MySQL y las que el desarrollador vaya ocupando en su labor. Ir a la opción de "Nueva".



- En esta nueva ventana, tendremos dos opciones para continuar, el nombre de la base de datos y el cotejamiento (Codificación de datos). Por defecto serán los siguientes serán





Este es el resultado final del proceso de crear la base de datos. Es muy sencillo cómo viste. En el próximo módulo realizamos la creación de las tablas de la base de datos y todos los temas que girarán en torno al proyecto, al PHP y MySQL.

Recursos disponibles para el aprendizaje



Para desarrollar las habilidades y destrezas necesarias en cada competencia, es muy importante que tengas acceso a los recursos didácticos adecuados.

Entonces, si necesitas reforzar esta información, te sugerimos revisar nuevamente los **Vídeos de Apoyos**, disponibles en el campus virtual, indicados en las lecturas anteriores. Además, recuerda que puede consultar las **Fuentes Documentales** que aparecen en esta guía, particularemnte, en el apartado de Referencias Bibliográficas.

MATERIAL COMPLEMENTARIO

Es importante continuar adquiriendo conocimiento y no frenar el proceso de aprendizaje, por esto es importante complementar lo aprendido en esta guía con nuevos conceptos, definiciones y caracteristica, sugerimos revisar el siguiente material:

HTML está presente en absolutamente toda la web y es importante tener presente todas y cada una de las etiquetas de forma global y básica. Te invito a estudiar y repasar un poco todas las etiquetas y características que componen este lenguaje de etiquetadas. Disponible en: https://www.w3schools.com/html/

La historia de HTML es interesante y engloba una cantidad muy significativa de versiones, carácterísticas y componentes para ser el concepto que es hoy. Conoce un poco de su historia. Disponible en: https://www.youtube.com/watch?v=B5VjL2f1IQg

Las bases de datos son las fuentes de cualquier software que se desarrolle, por esto te invito a leer un poco sobre este concepto. Disponible en: https://es.wikipedia.org/wiki/Base_de_datos

ASPECTOS CLAVES

- HTML está en todos los sitios web.
- PHP se integra perfectamente con HTML y sus etiquetas.
- PhpMyAdmin y MySQL viene integrado en XAMPP.
- Los métodos de envío de datos de los formularios son POST y GET.
- Los inputs tiene subcategorias para entradas diferentes de datos.
- Las tablas en HTML permiten definir una estructura organizada de información.
- Los textarea son entradas de texto varias líneas.
- MySQL será el motor de bases de datos que utilizaremos.
- MySQL permite realizar cualquier tipo de operación sobre bases de datos.

¡Felicidades! 👍 Has concluido con la lectura de la Guía Didáctica N°4. Así que ya puedes realizar la Evaluación 4.

Esta guía fue elaborada para ser utilizada con fines didácticos como material de consulta de los participantes en el Diplomado Virtual en Programación en PHP del Politécnico de Colombia, especialmente, a los técnicos, tecnólogos y profesionales de carreras afines, estudiantes de todas las carreras, empíricos, y público en general con conocimientos básicos en informática que intentan entrar en el mundo de la programación, que se desempeñen o no en las áreas de TIC de cualquier tipo de organización y que deseen obtener las competencias y habilidades necesarias para conocer los fundamentos prácticos del lenguaje de programación PHP para la aplicación y desarrollo de algoritmos y aplicaciones, y solo podrá ser reproducida con esos fines. Por lo tanto, se agradece a los usuarios referirla en los escritos donde se utilice la información que aquí se presenta.

Derechos reservados - POLITÉCNICO DE COLOMBIA, 2019
Medellín, Colombia